bajtek

Z MIKROKOMPUTEREM NA TY

NR 3 Cena 60 zł
LISTOPAD — 85

- Szanuj swój komputer
- Dźwięk w MERITUM
- Coś dla piratów
- Micro-PROLOG i LOGO cd.
- Magnetofon dla Commodore
- Sabre Wulf
- Flowchart
- Kluby komputerowe
- Olimpiada informatyczna



SŁOWA I FAKTY

Coraz głośniej znowu w Polsce o komputerach. Nie tylko w prasie. Oto tematem wykładu inauguracyjnego w szacownym Uniwersytecie Warszawskim była, uwaga: rewolucja mikroelektroniki. Nieczęsto można było dotychczas w Audytorium Maximum usłyszeć takie słowa jak te, które wypowiedział prof. Jacek Baranowski „Człowiek XXI wieku, nie umiejący posługiwać się komputerem, będzie się prawdopodobnie czuł jak analfabeta w wieku XX. Zastanawiający jest entuzjazm młodego pokolenia, z jakim chłonie ono wiedzę o programowaniu, i jak chętnie bawi się czy pracuje z komputerem. Tak jakby jakieś podświadome przeczucie podpowiadało tym młodym ludziom, że wiedza ta będzie im w najbliższym czasie najbardziej potrzebna".

Brawo, profesorze! Tak to właśnie zespół BAJTKA i nasi czytelnicy widzą. I, proszę nam wierzyć, naprawdę nie ma w tym niczego zastanawiającego!

To, że coraz więcej *m ó w i s i ę* o komputerach, musi cieszyć choć, oczywiście, cieszylibyśmy się jeszcze bardziej, gdybyśmy wreszcie mogli pisać również o tym, co się w tej dziedzinie *d z i e j e*. Krzemowej Doliny nie stworzymy, gdyż mogła ona powstać tylko raz – i to też przy szczególnie korzystnym zbiegu okoliczności. Możemy jednak zacząć realizować przemyślaną, na miarę możliwości – bo ambicje mamy póki co większe – koncepcję komputeryzacji gospodarki i edukacji informatycznej społeczeństwa. Są na to szanse – pokładamy je m.in. w prężnie rozwijającym swą działalność Urzędzie ds. Postępu Naukowo-Technicznego i Wdrożeń. To od niego zależy w dużym stopniu czy wczorajsze dyskusje przerodzą się w dzisiejsze decyzje i jutrzejsze fakty. Pokolenie wychowane na BAJTKU zacznie już wkrótce rozliczać z owych faktów tych, którzy wychowali się na „Swierszczyku". I to zapewne bez taryfy ulgowej...

Na koniec coś z własnego poletka. Miło mi otóż poinformować czytelników, że BAJTEK zaczyna robić (puk! puk! nie zapeszyć!) furorę międzynarodową. Okazuje się, po raz kolejny, że informacja nie zna granic. Sądząc po listach, jakie dostaliśmy z NRD, Czechosłowacji i Bułgarii, wiadomość o podjęciu wydawania BAJTKA i pierwsze egzemplarze naszego miesięcznika szybko tam dotarły. Trudno byłoby udawać, że się z tego nie cieszymy. Sam pokazałem niedawno kilka egzemplarzy pierwszych numerów BAJTKA kolegom w Moskwie. Nie mogłem im zrobić lepszego prezentu!

Czyżbyśmy więc – zapytam nieskromnie – trafili z BAJTKIEM w „dziesiątkę"?

WALDEMAR SIWIŃSKI



BAJTEK — dodatek specjalny „Sztandaru Młodych" i „Odrodzenia". Wydawca: Krajowe Wydawnictwo Czasopism RSW „Prasa-Książka-Ruch", ul. Noakowskiego 14, 00-666 Warszawa, tel. 25-72-94. Cena 60 zł Zam. 2019. N-13

SZANUJ SWÓJ KOMPUTER!

Duże komputery, zwłaszcza starszych typów, wymagały klimatyzacji, stałej temperatury otoczenia, specjalnej ochrony przed niepożądanymi wibracjami itp. Procesor w naszym Spectrum jest układem scalonym o wymiarach ok. 15 × 40 mm, może pracować w temperaturze od 0 do 70°C i drgania nie są dla niego aż tak zabójcze, ale... właśnie: komputer jest cierpliwy, zniesie bardzo wiele, jest wszak projektowany z uwzględnieniem zasady, którą Anglosasi z właściwym sobie poczuciem humoru nazwali „foolproof", czyli „idiotoodporność", ale wszystkiego nie zniesie i będzie protestował. Jest on jednak maszyną niezdolną do spontanicznego działania i jego protest może przybrać tylko jedną formę: potraktowany zbyt brutalnie zepsuje się.

Znamy pogardę, z jaką powszechnie traktuje się wszelkiego rodzaju instrukcje obsługi sprzętu elektronicznego, zwłaszcza jeśli są one sporządzone w językach obcych, nie zawsze znanych użytkownikowi. Zebraliśmy więc tu podstawowe informacje, które pozwolą Wam uniknąć płacenia zbyt wysokiej ceny za doświadczenia zbierane metodą prób i błędów, popełnianych często już przy pierwszym włączeniu urządzenia.

Zacznijmy od norm. Komputery projektowane są z myślą o pracy w pomieszczeniach zamkniętych w temperaturze od 10 do ok. 40 st. C przy względnej wilgotności powietrza nie przekraczającej 80 proc. — zupełnie suche powietrze też nie jest najlepsze, gdyż sprzyja gromadzeniu się elektryczności statycznej, która zakłóca pracę maszyny, a ewentualny przeskok iskry może spowodować nawet jej całkowite zniszczenie. Przepływ bardzo suchego gorącego powietrza związany z intensywnym grzaniem się pewnych elementów (ULA, stabilizator w zasilaczu itp.) zwłaszcza że firma Sinclair przez długi czas mowych elementów klawiatury, zwłaszcza folii, z której zrobione są zestyki, a także źle wpływa na obudowy układów scalonych, których korozja (zmiany struktury polimerów), choć bardzo powolna, po kilkunastoletnim używaniu sprzętu stanie się istotnym czynnikiem wpływającym na jego trwałość. Optymalny zakres wilgotności to 20—40 proc., co w praktyce oznacza pokój z lekko uchylonym oknem lub w inny sposób zapewnioną wentylacją, a w zimie — z prostym nawilżaczem na kaloryferze lub piecu. Wymagania komputera co do wilgotności i temperatury są zresztą bardzo podobne do naszych: tam gdzie człowieka drażni wilgoć lub zbyt suche powietrze (choć w tym wypadku trudniej skojarzyć sobie objawy złego samopoczucia z przyczyną) — tam i maszyna będzie cierpiała.

Komputer nie lubi kurzu, choć nie zawiera przecież żadnych części mechanicznych, które mogłyby się zatrzeć, a w klawiaturze zestyki z folii są dobrze chronione przed zabrudzeniem. Jeśli jednak kurzu zgromadzi się zbyt dużo, to po zawilgoceniu może nastąpić zwarcie ścieżek układu scalonego. Grozi to zniszczeniem elementów, których koszt jest porównywalny z ceną nowego urządzenia.

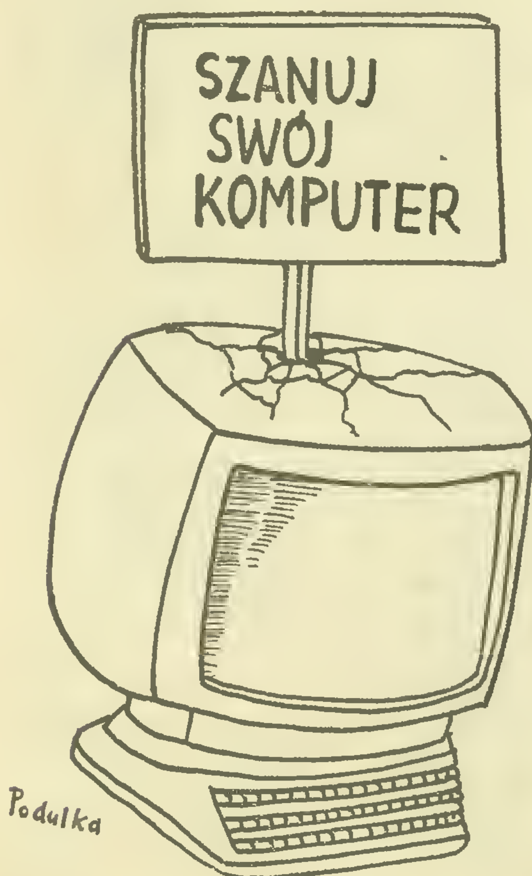
Podsumujmy: komputer będzie świetnie się czuł i wiernie nam służył, jeśli nie wystawimy go na słońce lub deszcz, nie będziemy nosić na plażę lub zakurzone podwórze, a po pracy włożymy do pudełka lub przykryjemy czymkolwiek, co uchroni go przed pyłem. Nie wolno natomiast przykrywać go podczas pracy ani też ustawiać na miękkim podłożu, które utrudni dostęp powietrza do otworów wentylacyjnych, umieszczonych na spodzie obudowy — ZX Spectrum jest szczególnie wrażliwy na nadmierny wzrost temperatury w wyniku pogorszenia się chłodzenia podczas pracy.

Przegrzanie komputera powoduje zakłócenia w pracy procesora, który z początku zaczyna robić rozmaite dziwne rzeczy, czego efekty widać na ekranie TV. Podobne efekty powoduje przegrzanie się ULA, układu odpowiedzialnego za współpracę komputera z urządzeniami zewnętrznymi. Jest to wysoce specjalizowany układ scalony, zaprojektowany i produkowany wyłącznie dla firmy Sinclair. Jest więc bardzo drogi, zwłaszcza że firma Sinclair przez długi czas utrudniała jego zakup bez komputera, chroniąc

się w ten sposób przed próbami kopiowania Spectrum. Obecnie ULA kosztuje ok. 20 dol., a na „Perskim Jarmarku" w Warszawie — powyżej 20 tys. zł. Niebezpieczeństwo nadmiernego wzrostu temperatury występuje zwłaszcza w wersji 48kB. Wada ta została w znacznym stopniu wyeliminowana w Spectrum Plus. w wyniku dużo korzystniejszego ukształtowania obudowy i rozmieszczenia otworów chłodzących.

Ważnym — choć z reguły niedocenianym — elementem składowym komputera jest zasilacz. W Spectrum składa się on z dwóch zasadniczych bloków funkcjonalnych: transformatora sieciowego zamkniętego w oddzielnej obudowie wraz z prostownikiem i filtrem tętnień oraz stabilizatora i powielacza wewnątrz obudowy komputera.

Wbrew pozorom pudełko z transformatorem jest istotnym elementem całego urządzenia. Od jakości jego elementów i dostarczanego przez nie napięcia zależy poprawna praca całego urządzenia. Szczególnie ważna jest filtracja tętnień i zakłóceń występujących w sieci. Zdarzają się niestety fabryczne usterki polegające np. na... braku kondensatora filtru tętnień. Przyłączenie takiego zasilacza do komputera musi spowodować jego uszkodzenie.



Dla czytelników obeznanych nieco z elektroniką podajemy, że napięcie dostarczane przez sprawny zasilacz bez obciążenia wynosi od 13 5 do 14 5 V, a pod obciążeniem spada do ok. 12—13 V. Tętnienia sieci na wyjściu nie powinny być większe niż 2 V. Niektórzy użytkownicy próbują eksploatować Spectrum przy pomocy zasilaczy o napięciu obniżonym do 11—12 V — bez obciążenia i 1—10 V — pod obciążeniem. Rozwiązanie takie poważnie zmniejsza straty energii w stabilizatorze, a więc i grzanie się całego komputera, obniża jednak pewność jego pracy: spadek napięcia zasilającego poniżej ok. 7 V powoduje już błędy w pracy procesora i pamięci, a także zwiększa ich podatność na trwałe uszkodzenia.

Drugim, a zarazem najważniejszym blokiem zasilacza jest jego część znajdująca się wewnątrz obudowy komputera, zmontowana na wspólnej z nim płytce. Dostarcza ona stabilizowanych napięć potrzebnych do zasilania pamięci RAM 16K. Niestety jest to także słaby element naszego Spectrum. Przetwornica wytwarzająca napięcie 12 V pracuje na krawędzi swych możliwości i zdecydowana większość awarii komputera rozpoczyna się od uszkodzenia tego własnie elementu. Piszemy rozpoczyna się, gdyż prawie nigdy na tym się nie kończy: związany z przepaleniem się impuls przetwornicy powoduje zwykle uszkodzenie ULA, kilku kości pamięci RAM, zdarzają się też uszkodzenia ROM i mikroprocesora.

Niestety, do spowodowania tego rodzaju uszkodzenia wystarcza bardzo niewiele: jakiekolwiek przypadkowe zwarcie szyn komputera, np. w wyniku dotknięcia metalowym przedmiotem (długopisem, monetą, spinaczem itp.) do niczym nie osłoniętej szyny złącza krawędziowego, wyprowadzonego z tyłu komputera. Z drugiej strony otworu tego podczas pracy komputera nie należy zasłaniać, gdyż w ten sposob ograniczamy obieg powietrza chłodzącego. Najlepszą ochroną byłaby cienka nakładka chroniąca złącze, a nie utrudniająca przepływu powietrza, trudno jednak coś takiego wykonać w warunkach amatorskich.

Przyczyną awarii przetwornicy może być też niedokładne włożenie wtyczki sieciowej do gniazda, a nawet impuls zakłócający powstały w wyniku włączenia do sieci źle odkłóconego urządzenia dużej mocy, np. pralki lub grzejnika. Bardzo groźne dla Spectrum są też chwilowe zaniki napięcia sieciowego powodujące tzw. przepięcia. Nawet zwykłe krótkotrwałe i szybko następujące po sobie spadki napięcia o ponad 10 proc. mogą wyczerpać cierpliwość maszyny (wcześniej pisaliśmy o próbach trwałego zasilania niższym napięciem, a nie o nagłych jego zmianach).

Jak uchronić nasz komputer przed uszkodzeniami tego typu? Bardzo pomocny jest tu sta-



bilizator napięcia sieci, podobny do stosowanego w odbiornikach TV. Zapewnia on dostatecznie stabilne napięcie i eliminuje poważną część zakłóceń sieciowych.

Pamiętajmy też, że wspaniałe pomysły w postaci wyłączników instalowanych na kablu między komputerem a zasilaczem przynoszą z reguły więcej strat niż pożytku, nie zapewniają bowiem na ogół odpowiedniej jakości styku i mogą powodować iskrzenie. Czym natomiast kończy się zwykle iskrzenie, już wiemy.

Wilgoć nie jest wprawdzie aż tak niebezpieczna jak niestabilność sieci, ale może ona spowodować zakłócenia uniemożliwiające (co najmniej chwilowo) pracę komputera, a występując ciągle może wywołać wadliwą pracę klawiatury, aż do trwałego zniszczenia niektórych zestyków.

Klawiatura ZX Spectrum nie jest niestety najlepszej jakości, wiele do życzenia pozostawia m.in. jej trwałość. Jeżeli często bawimy się grami zręcznościowymi (typu ARCADE), lepiej wydać nieco pieniędzy i kupić manipulator (joystick) oraz interface. Na dłuższą metę jest to bardziej opłacalne niż naprawa lub wymiana klawiatury.

Jeśli wykorzystujemy nasz komputer do pracy, lepiej kupić dodatkową klawiaturę skonstruowaną z zestyków kontaktronowych, oferowanych przez kilka firm produkujących dodatkowe wyposażenie do komputerów, lub po prostu kupić — co jest o wiele tańsze — nową obudowę i przerobić nasz komputer na Spectrum plus, którego klawiatura jest już dużo lepsza, choć w niej z kolei częściej zdarzają się zwarcia, związane z wielowarstwowością folii i bardziej skomplikowanym przebiegiem ścieżek.

Na zakończenie powtórzmy kilka podstawowych zasad pracy z komputerem, zapewniających urządzeniu, a w konsekwencji i nam, dobre samopoczucie:

— wszystkie manipulacje, takie jak przyłączanie i wyłączanie dodatkowych urządzeń peryferyjnych, wykonujemy przy wyłączonym komputerze, zwłaszcza jeśli wykorzystujemy w tym celu złącze krawędziowe.

— przed przyłączeniem czegokolwiek do złącza upewniamy się, czy nasz wtyk wyposażony jest w plastykową wstawkę wskazującą (tzw. slot) a więc wycięcie w płytce komputera między 4 a 5 parą styków. W wycięciu tym czasami pozostają resztki ułamanej w wyniku wyszarpywania wtyku wstawki i wówczas łatwo o niemiłą w skutkach pomyłkę. Wtyk z wypadniętą wstawką aż zachęca do błędu, który kończy się zwykle wymianą procesora i np. pamięci ROM. Szczególnie podatne na tego rodzaju uszkodzenia są wtyki drukarek Seikoscha GP-50.

— z wszelkich połączeń eliminujemy natychmiast po zauważeniu niepewne styki, nieizolowane lub niekontaktujące kable itd.

— zasilacz najpierw włączamy do sieci a później do komputera

— unikamy przegrzewania komputera

— z miejsca pracy usuwamy wszystkie przedmioty metalowe, które mogą dostać się do środka przez szczelinę złącza krawędziowego

— jeśli w miejscu naszego zamieszkania zdarzają się częste wyłączenia prądu lub napięcie sieci jest niestabilne, kupujemy stabilizator.

Życzę przyjemnej pracy i zabawy bez żadnych awarii.

GRZEGORZ GRZYBOWSKI

Zdaniem wielu użytkowników brak dźwięku jest jednym z poważniejszych niedostatków mikrokomputera MERITUM. Jest to jednak wada pozorna: wszystkie MERITUM, począwszy od pierwszych egzemplarzy, posiadają możliwość uzyskiwania efektów dźwiękowych, nie było to jednak opisywane w oficjalnych publikacjach producenta.

DŹWIĘK W MERITUM I



Generowanie sygnału dźwiękowego odbywa się poprzez podawanie do przerzutnika umieszczonego pod adresem wejścia (wyjścia 254) (FE w zapisie heksadecymalnym, czyli szesnastkowym) w odpowiednich odstępach czasu bajtów, których ostatni, najmniej znaczący bit ma wartość na przemian 0 i 1. Sygnał o odpowiedniej częstotliwości jest więc generowany programowo.

We wcześniejszych egzemplarzach MERITUM wzmacniacz i głośnik znajdowały się w mikrokomputerze, natomiast w egzemplarzach późniejszych wykorzystuje się tor fonii monitora Neptun 156 współpracującego z mikrokomputerem.

Najprostszym sposobem uzyskania dźwięku jst uruchomienie odpowiedniego programu w BASIC-u:

```
10 OUT 254, 0: OUT 254, 1 : GOTO 10
```

lub

```
FOR n=1 TO 2 STEP 0 : OUT 254, 0 : OUT 254, 1: NEXT
```

Programy te mają jednak wartość wyłącznie poglądową, uzyskana podczas ich wykonywania częstotliwość sygnału jest rzędu 30 Hz (ze względu na niedostateczną szybkość działania interpretera języka BASIC-MERITUM). Drugi z podanych programów jest nieco szybszy.

Sygnał o wyższej częstotliwości można uzyskać wykorzystując odpowiednią procedurę napisaną w języku wewnętrznym mikroprocesora Z-80.

Oto najprostsza tego typu procedura generująca sygnał o częstości 500 Hz:

```
LD D, 7D      pętla opóźniająca. Czas jej wy-
              konywania, a więc częstotliwość
LD B, D       sygnału można regulować zmie-
DJNZ — 2      niając wartość (7D hex) wpisy-
              waną w pierwszej linii do reje-
              stru D
LD A,1
OUT (FE), A   wysyła 1 na wyjście 254
LD B, D
DJNZ — 2      pętla opóźniająca
LD A, 0
OUT (FE), A   wysyła 0 na wyjście 254
IR — 16       powrót do początku programu
```

Procedurę tę można skrócić wykorzystując fakt, że istotny jest tylko najmniej znaczący

bit podawanego na wyjście 254 bajtu, a więc wystarczy wysyłać na zmianę liczby parzyste i nieparzyste:

```
LD D, 7D
LD B, D
DJNZ — 2
INC A          zwiększa zawartość akumulatora o 1
OUT (FE), A
JR — 8
```

Program ten można wpisać do pamięci i wykonać z poziomu BASIC-u następująco:

10 POKE 16526,0 : POKE 16527,125 podaje interpretorowi adres startowy procedury

20 FOR P=32000 TO 32009: READ S: POKE P, S: NEXT P czyta dane i umieszcza je pod właściwymi adresami

30 DATA 22, 125, 66, 16, 254, 211, 254, 24, 248
40 U=USR (0)

W MERITUM program w języku wewnętrznym można wpisać również korzystając z umieszczonego pod adresem 3000 hex (12288 dziesiętnie) monitora. Opis działania tego monitora podamy wkrótce, a obecnie dla umiejących się nim posługiwać podajemy tekst powyższego programu w zapisie heksadecymalnym: 16, 7D, 42, 10, FE, 3D, D3, FE, 18, F8

Program ten nie przewiduje powrotu z procedury w języku wewnętrznym do BASIC-u, powrot taki jest więc możliwy jedynie w wyniku naciśnięcia klawisza NMI. Liczy on jedynie 10 bajtów, istnieje jednak jeszcze krótszy program — 9 bajtowy, który powoduje generowanie dźwięku pulsującego typu „syrena strażacka" — aby go uzyskać, wystarczy w podanym programie zastąpić pierwsze dwa bajty (LD D, 7D) przez jeden — INC D, czyli 20 w zapisie dziesiętnym lub 14 w zapisie hex i zmienić ostatni bajt na F7, czyli 247 dziesiętnie. W podanym wyżej programie w BASIC-u należy w tym celu linię 30 zastąpić linią 30 DATA 20, 66, 16, 254, 60, 211, 254, 24, 247, 0

Rozszerzając ten program uzyskujemy możliwość wytwarzania dźwięków o zadanej wysokości i czasie trwania. Oto przykładowe rozwiązanie wykorzystujące nieco bardziej skomplikowane konstrukcje, podajemy je więc bez szczegółowego komentarza:

```
LD de, NN
dwa bajty wskazujące długość dźwięku
INC a
OUT FE
PUSH af
LD b, N
DEC de
LD a, d        ta sekwencja oznacza
OR e           sprawdzenie czy de=0
JR Z+5
PJNZ—7
POP af
JR — 16
POP af
RET
```

Tak wygląda ten program (21 bajtów) w zapisie hex:

11, X, X, 3C, D3, FE, F5, 6, X,1B, 7A, B3, 28, 05, 10, F9, F1, 18, F0, F1, C9
(X oznacza wartość podawaną dowolnie przez użytkownika)
w zapisie dziesiętnym:
17, x, x, 60, 211, 254, 245, 6, x, 27, 122, 179, 40, 5, 16, 249, 241, 24, 240, 241, 201.

Wartości bajtów regulujących długość i wysokość dźwięku (oznaczonych przez x) należy wprowadzić z poziomu BASIC-u poprzez POKE, przy czym drugi bajt jest bardziej znaczący, tzn. na przykład wartość 2000 zapisujemy jako 256 razy 7 (starszy bajt) plus 208 (młodszy bajt), np.:

```
5 CLEAR0
10 POKE 16526,0 : POKE 16527,125
20 FOR P=32000 TO 32020 : READS: POKE P,S: NEXTP
30 DATA 17, 0, 0, 60, 211, 254, 245, 6, 0, 27, 122, 179, 40, 5, 16, 249, 241, 24, 240, 241, 201
40 INPUT "WYSOKOŚĆ"; H: INPUT "DŁUGOŚĆ (DO 255)"; L
50 POKE 32002, L : POKE 32008, H
60 PRINT USR (0): GO TO 40
```

MACIEJ DEPTUCH
WŁADYSŁAW MAJEWSKI

COŚ DLA PIRATÓW

Z MAGNETOFONU NA MAGNETOFON

Oto najkrótszy program kopiujący ze Spectrum, któremu nic się nie oprze — kopiuje nawet muzykę, nawet programy firmy US GOLD chronione trikiem zawartym w programie ładującym, i powodującym, że właściwy program nagrany jest z inną prędkością, a sygnał pilotujący przypomina rozruch traktora, tak więc przez normalne programy kopiujące w ogóle nie jest rozpoznawany jako program!

Wystarczy po prostu jeden magnetofon z taśmą oryginalną podłączyć do gniazdka EAR, tak jak byśmy chcieli ładować program do magnetofonu, drugi z taśmą, na której chcemy mieć kopię, podłączamy do gniazdka MIC — tak jakbyśmy chcieli nagrać program. Jakość nagrania będzie nawet lepsza niż na taśmie oryginalnej: program zdąży z dostateczną prędkością odczytać stan wejścia, zinterpretować jako sygnał cyfrowy i już jako taki wysłać na wyjście.

Przed rozpoczęciem przegrywania piszemy:

```
10 FOR n =USR "a" TO USR "a"+8
20 READ a : POKE n,a: NEXT n
30 DATA 243, 14, 254, 237, 120, 237, 121, 24, 250
40 RANDOMIZE USR USR "a"
```

Jak widać, cały nasz program liczy zaledwie 9 bajtów kodu maszynowego. Oto znaczenie kodów podanych w instrukcji DATA:

243	di	wyłącz przerwania — przegrywanie nie może być przerywane 50 razy na sekundę na czas potrzebny komputerowi do odczytania klawiatury
14 254	ld c, 254	wprowadź wartość 254 do rejestru c — 254 jest to numer portu, który w Spectrum oznacza wejście i wyjście magnetofonowe
237 120	START in a, (c)	wczytaj z portu wskazanego przez zawartość rejestru c, czyli z magnetofonu, bajt danych do akumulatora
237 121	out (c), a	poślij ten bajt z powrotem na magnetofon
24 250	JR START	wróć do punktu START (czyli 6 bajtów wstecz)

Program ten działa bez zarzutu i bardzo skutecznie, ma tylko jedną wadę — przerwać go możemy tylko wyciągając sznur zasilający z komputera lub wykonując (w Spectrum) — RESET.

Uwaga! Podczas przegrywania przy pomocy tego programu ekran stale jest pusty i BORDER nie miga!

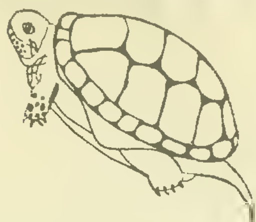
INTERFACE III

Dla tych, którzy mają pieniądze, nie mają natomiast czasu i męczy ich przegrywanie programów opisaną wyżej metodą: bo i samo przegrywanie trwa, i potem jeszcze trzeba sprawdzić, co się przegrało, firma Evasham Micro ma za 40 funtów ofertę nie od odrzucenia: Interface III kopiuje na Microdrivie wszystko, co znajdzie się w komputerze — niezależnie od tego, jak zabezpieczone.

Wystarczy po prostu nacisnąć guziczek — i Interfejs przegra całą zawartość pamięci Spectrum na kasetkę Microdriv-u. Producenci oprogramowania postulują skazanie właścicieli i dyrekcji z firmy Evasham na 20 lat Tower — takie kary za piractwo komputerowe przewiduje od niedawna ustawodawstwo brytyjskie. W świetle ustawy podpisanej przez królową produkowanie i sprzedawanie takich urządzeń niczym nie różni się od produkowania i sprzedawania uniwersalnych wytrychów.

Nawiasem mówiąc, ustawa ta przewiduje wysokie grzywny za samo posiadanie skopiowanego nielegalnie oprogramowania! Firma Evasham zdążyła wprowadzić na rynek swoje urządzenia przed wejściem nowej ustawy — i solidnie na tym zarobić, bo klienci kupowali na wyścigi! Oficjalnie firma głosi zresztą, że jej urządzenie służy do sporządzania zapasowych kopii na własny użytek...

(YOUR COMP. JULY 85)



LOGO LOGO LOGO

Kończymy dziś podstawowy opis LOGO.

REPEAT

Instrukcja REPEAT (powtórz) umożliwia wykonanie fragmentu programu zadaną z góry liczbę razy, np. REPEAT 4 [TYPE "BAJTEK] daje BAJTEKBAJTEKBAJTEKBAJTEK — pierwszy parametr to liczba powtórzeń, drugi to lista czynności do powtarzania.

IF

Często chcemy, by fragment programu wykonywany był tylko w przypadku spełnienia pewnego warunku. Stosujemy wówczas konstrukcję IF warunek [lista czynności wykonywanych w razie spełnienia warunku] oraz ewentualnie [lista czynności wykonywanych w razie niespełnienia warunku]. Warunek musi przyjmować wartości TRUE lub FALSE (prawda lub fałsz), może więc być wyrażeniem logicznym, zmienną logiczną lub procedurą dającą wartość logiczną na wyjściu. Przykład:

```
TO WYBOR
OP IF 0 = RANDOM 3 ["TAK] ["NIE]
END
PR WYBOR
TAK
```

AND
OR
NOT

pokazuje, że IF może wybierać nie tylko czynności, ale i przekazywane procedurze nadrzędnej listy lub wartości.
Wyrażenie logiczne tworzy się przy pomocy operatorów: **AND** (i), **OR** (lub) lub **NOT** (zaprzeczenie), np.

```
PR (AND 16 = 16 NOT 2 = 3 (OR 6 > 5 5 > .6 5 = 6))
TRUE
```

Jak widać, instrukcje OR i AND są zachłanne. Wyrażeniami logicznymi lub ich częściami mogą być też pytania LOGO, np. **WORDP, LISTP, DEFINEDP itp.**

RUN

W pierwszym odcinku omówiliśmy możliwość tworzenia programów samomodyfikujących drogą budowy wewnątrz programu nowych definicji. Istnieje też prostszy sposób: RUN lista czynności powoduje wykonanie tej listy.

ROZMOWA Z KOMPUTEREM

W trakcie wykonywania programu komputer może sprawdzać, czy jest wciśnięty jakikolwiek klawisz. Jeśli jest, wów-

LOGO LOGO LOGO

KEYP
READCHAR RC

czas KEYP = 1. Jeśli interesuje nas, co to za klawisz, możemy posłużyć się operacją READCHAR (RC). Komputer czeka na wciśnięcie klawisza (tak jak w BASIC-u INPUT), ale nie wymaga naciśnięcia ENTER i nie wyświetla na ekranie przyjętego znaku (jak w BASIC-u INKEY). Można wprowadzić również całą listę przy pomocy zlecenia READLIST (RL). Oczywiście w tym wypadku konieczne jest zaznaczenie końca wprowadzanego tekstu klawiszem ENTER.

READLIST RL

PRINT PR
SHOW
TYPE

Komputer wypowiada się na zlecenie PRINT (PR), SHOW i TYPE. Pierwsze dwa powodują przejście do następnej linii po wykonaniu zlecenia, natomiast TYPE powoduje pozostanie w dotychczasowym miejscu. SHOW wyświetla listy razem z obejmującymi je nawiasami. Miejsce, w którym będzie ukazywał się tekst, określa położenie kursora (znacznika). Ustalić je można przy pomocy operacji CURSOR. która zwraca listę zawierającą numer znaku w linii i numer linii jako listę. Zleceniem SETCURSOR (SETCUR) [a b] można ustawić kursor w miejscu znaku a kolumny b.

CURSOR
SETCURSOR
SETCUR

SETTC

Kolor napisów i ich tła można ustawić zleceniem SETTC [napis tło], np. SETTC [2 4] oznacza napis czerwony na zielonym tle .x — w odróżnieniu od SETBG instrukcja SETTC ustala kolor tła nie dla całego ekranu, a wyłącznie dla stawianych na nim znaków.

FLASH
INVERSE

NORMAL
OVER

BRIGHT

Podobnie działają instrukcje FLASH i INVERSE, których znaczenie jest analogiczne do ich znaczenia w ZX Spectrum — BASIC. Powrót do normalnych atrybutów umożliwia instrukcja NORMAL. OVER 1 powoduje, że znaki tekstu nie niszczą tekstu lub rysunku znajdującego się w miejscu ich wyświetlania. Powrót do stanu normalnego poprzez OVER 0. Analogicznie funkcjonuje BRIGHT 1 i 0 — włączanie i wyłączanie rozjaśnionego tła.

CLEARTEXT CT

TEXTSCREEN TS

SOUND

Instrukcja CLEARTEXT (CT) usuwa wszystkie teksty z ekranu — w trybie graficznym z dwóch dolnych linii, a instrukcja TEXTSCREEN (TS) przeznacza cały ekran na teksty, usuwając żółwia i jego rysunki. Komputer może się też wypowiadać na głos — przy pomocy instrukcji SOUND czas trwania (0—255) wysokość (—62 do 75).

MAGNETOFON

SAVE
SAVEALL
SAVED
SAVESCR

Wszystko, co stworzyliśmy w LOGO, możemy zapisać na taśmę przy pomocy SAVE "nazwapliku [lista procedur]. Całą zawartość pamięci można nagrać na taśmę przy pomocy instrukcji SAVEALL, a zawartość edytora — SAVED. Obrazek z ekranu można nagrać przy pomocy SAVESCR.

LOGO LOGO LOGO

SETDRIVE
CATALOG
ERASEFILE

Instrukcje SETDRIVE, CATALOG, ERASEFILE interesują wyłącznie posiadaczy microdrive'ów, ich znaczenie jest oczywiste.

LOAD
LOADD
LOADSCR

To, co uprzednio nagraliśmy, możemy ponownie wprowadzić do komputera przy pomocy LOAD "nazwapliku. LOADD i LOADSCR służą odpowiednio do ładowania zawartości edytora i ekranu.

DRUKARKA

PRINTON
PRINTOFF

Instrukcja PRINTON powoduje, że wszystkie teksty wysyłane przez LOGO na ekran są równocześnie drukowane. PRINTOFF wyłącza drukarkę.

COPYSCREEN

Istnieje też możliwość skopiowania ekranu — następuje to po instrukcji COPYSCREEN.

SYSTEM

.DEPOSIT
.EXAMINE
.CALL

Tworząc programy w LOGO można posługiwać się procedurami napisanymi w języku wewnętrznym. Rolę podobną do POKE w BASIC-u pełni instrukcja .DEPOSIT adress n — umieszcza ona pod zadanym adresem pamięci podaną wartość. Odpowiednikiem PEEK jest .EXAMINE adres (odczytaj zawartość podanej komórki), a odpowiednikiem USR jest .CALL adress — woła procedurę umieszczoną pod podanym adresem.

.BSAVE
.BLOAD
.SETSERIAL n
.SERIALIN
.SERIALOUT

Przy pomocy .BSAVE „nazwapliku [adrespoczątkowy] długość można nagrać na taśmę zawartość pamięci — jako bajty. Odczyt następuje przy pomocy .BLOAD "nazwapliku adres. Instrukcja .SETSERIAL umożliwia korzystanie z portu szeregowego do komunikacji komputera z otoczeniem przy zadanej prędkości transmisji n (w bodach, od 50 do 19200). .SERIALIN i .SERIALOUT umożliwiają odpowiednie odczytanie danych z portu i wysłanie n bajtów.

.RECYCLE
.NODES
.CONTENTS

LOGO gospodaruje pamięcią w sposób dynamiczny i rzeczy zbędne usuwa, gdy zaczyna brakować miejsca. Wówczas żółw zatrzymuje się na 2—3 sek. i "sprząta po sobie". Można nakazać sprzątanie w wybranym przez siebie momencie, co poprawia rytm wykonywania programu. Wykonuje to instrukcja .RECYCLE. Bezpośrednio po .RECYCLE można przy pomocy operacji .NODES uzyskać informację, ile jeszcze jest wolnych tzw. nodów, czyli jednostek pamięci do 5 bajtów każda. .CONTENTS pozwala przejrzeć zawartość pamięci.

Uwaga — zlecenia, których nazwy rozpoczynają się od kropki, są rozszerzeniem LOGO wychodzącym poza strukturę i należy korzystać z nich b. ostrożnie.

W.M.

PROGRAMOWANIE W JĘZYKU LOGIKI PROLOG cz. 2

Prolog nie jest przystosowany do wykonywania złożonych obliczeń matematycznych, jednak wbudowane weń relacje mogą być przydatne do rozwiązywania niektórych problemów o charakterze czysto arytmetycznym.

Do podstawowych relacji arytmetycznych należą: **SUM, TIMES, LESS, INT.**

Mogą one przyjmować wartość 1 (prawda), 0 (fałsz).

Relacja SUM (x y z) jest prawdziwa jedynie wtedy, gdy z=x+y.

```
&. is (SUM (30 30 50))
YES
```

Pytanie o wynik dodawania formułujemy w następujący sposób:

```
&. which (x: SUM (20 30 x))
50
```

Wynik odejmowania np. (50—30) możemy otrzymać na trzy sposoby:

```
&. which (x: SUM (50 — 30 x))
20
```

lub:

```
&. which (x: SUM (30 x 50))
20
```

czy też:

```
&. which (x: SUM (x 30 50))
20
```

W relacji SUM może występować tylko jedna niewiadoma, czyli Prolog nie odpowie na pytanie:

```
&. which (x y : SUM (x y 10))
```

że rozwiązania to zbiór punktów na prostej o równaniu y=x—10, lecz zasygnalizuje błąd.

Too many variables (Zbyt dużo zmiennych)

Arytmetyka Prolog-u rozróżnia liczby całkowite i zmiennoprzecinkowe dodatnie oraz ujemne. I tak liczbami całkowitymi będą:

10 25343 — 65532 — 987 ...

a zmiennoprzecinkowymi:

5.9887 534E3 7.884646E32 ...

Relacja INT może służyć do sprawdzenia czy dana liczba jest całkowita lub zmiennoprzecinkowa bez części ułamkowej, oraz do wyznaczania całkowitej części liczby fp. floating point — zmiennoprzecinkowy.

Pytania formułujemy w sposób następujący:

```
&. is (45 INT)
YES
&. is (4 — 67 INT)
NO
&. is (3.567E3 INT)
YES
```

Natomiast przy wyznaczaniu części całkowitej piszemy:

```
&. which (x:3.45 INT x)
3
&. which (x:3.56398E3 INT x)
3563
```

Żeby sprawdzić czy jakaś liczba jest częścią całkowitą innej, możemy połączyć relację INT z EQ (od ang. EQual — równe)

Relacja TIMES ma następującą definicję: TIMES (x y z) zachodzi wtedy i tylko wtedy, gdy z=x · y. Relację TIMES użyć możemy (analogicznie jak SUM) na kilka sposobów:

```
&. is (TIMES (3 4 12))
YES
&. which (x: TIMES (3 4 x))
12
&. which (x: TIMES (3 x 12))
4
&. is (TIMES (3 y 12) & y INT)
YES
```

to ostatnie pytanie ma na celu sprawdzenie czy wynik dzielenia 12 przez 3 jest całkowity. Jeśli natomiast chcemy zrealizować dzielenie całkowite, piszemy:

```
&. which (x : TIMES (24 y 126) & x INT y)
5
```

ostatnie pytanie moglibyśmy zinterpretować następująco: „jaki x jest częścią całkowitą takiego y, że 24 · y = 126"

Relacja **LESS** może być używana jedynie do sprawdzania pewnych wyrażeń:

LESS (x y) zachodzi wtedy i tylko wtedy, gdy x jest mniejsze od y.

&. is (3 LESS 4) daje odpowiedź **YES**, gdyż 3 jest mniejsze od 4.

Podobnie na:

&. is (4 LESS 3) Prolog odpowiada:

```
NO
```

Również pytania:

```
&. is (TIMES (3 x 10) & TIMES (3 x y) &
1. SUM (y z 10) & z LESS 0.1E—5)
YES
```

(1. oznacza kontynuację poprzedniego wiersza). Prolog nie pozostawia bez odpowiedzi. Bardziej zaawansowanym miłośnikom micro-Prolog-u pozostawiam analizę semantyczną powyższego pytania.

LESS może również porównywać zmienne łańcuchowe, szeregując je alfabetycznie.

```
&. is (FRED LESS FREDDY)
YES
&. is (ALBERT LESS HAROLD)
YES
&. is (SAM LESS BILL)
NO
```

Przy formułowaniu pytań należy pamiętać o tym, że Prolog wszystkim wyrażeniom logicznym i arytmetycznym nadaje wartości kolejno od strony lewej do prawej. Dlatego na pytanie:

&. which (x : SUM (y 10 x) TIMES (2 5 y))

Prolog odpowiada:

Too many variables, zaś na analogiczne:

&. which (x : TIMES (2 5 y) & SUM (y 10 x))
29

☆

Interesujący jest sposób, w jaki Prolog odpowiada na zwykłe pytania: is (.....) gdzie "....." jest dowolnym zdaniem nie zawierającym zmiennych.

np. by znaleźć odpowiedź na pytanie:

&. is (Henry male)

Prolog wyszukuje wszystkie obiekty posiadające cechę male:

Henry-Snr male
Henry male
Edward male,

następnie przyrównuje Henry do pierwszego obiektu i jeśli są one równe, przechodzi do następnego.

Gdy znajduje obiekt Henry, to przekazuje wiadomość **YES**, w przeciwnym wypadku **NO**.

Gdy w takim pytaniu występuje zmienna, to Prolog najpierw stara się nadać jej jakąś wartość (liczbową lub literową), a cała dalsza procedura jest taka sama. Dlatego też przy rozbudowanych pytaniach jest do sprawdzenia bardzo wiele warunków i czas oczekiwania na odpowiedź się wydłuża.

W celu głębszego zrozumienia oraz prześledzenia etapów wartościowania każdego zdania (pytania lub stwierdzenia) można skorzystać z programu **SIMTRACE**. Wczytujemy go komendą

Load SIMTRACE

Blok **SIMTRACE** jest napisany w oryginalnym micro-Prologu (podobnie jak SIMPLE) i służy do śledzenia pracy systemu. Napiszmy:

&. all-trace x: Henry-Snr father-of x
1. and x male

Pierwszym wyrażeniem, którego wartość logiczną można zbadać, jest: Henry-Snr father-of x, dlatego **SIMTRACE** wypisuje wiadomość:

(1) Henry-Snr father-of x trace?

z zapytaniem, czy śledzić przebieg dobierania obiektów do x w celu uzyskania logicznej prawdy.

Jeżeli chcemy oglądać przebieg procesu wartościowania pytania, naciskamy „y" lub ENTER, zaś „n" w przeciwnym przypadku.

Jeśli naciśniemy "y" następna wiadomość wygląda następująco:

(1) solved : Henry-Snr father-of Mary

Prolog odnalazł obiekty Mary o własności:

Henry-Snr father-of Mary

Teraz SIMTRACE analizuje następny warunek i pisze:

(2) Mary male trace?

Po naciśnięciu „y" otrzymujemy

failing (2) i zaraz potem

failing (1)

Przyjrzyjmy się dokładnie działaniu **SIMTRACE:**

System rozpatrywał najpierw pytanie Henry-Snr father-of x. Pierwszym obiektem, znalezionym i spełniającym pierwszy warunek był X-Mary. Lecz następny warunek brzmiał: male. Prolog podstawił pod X Mary i uzyskał zdanie Mary male, nie znalazł go jednak w słowniku relacji, więc przyjął je za fałszywe. Blok **SIMTRACE** wysłał w tym momencie wiadomość o niespełnieniu drugiego warunku przez obiekt X-Mary stąd właśnie failing (2). Pozostało więc już tylko obliczenie koniunkcji dwóch zdań: prawdziwego i fałszywego, w wyniku którego **SIMTRACE** wysłał nową wiadomość: failing (1) oznaczającą niespełnienie koniunkcji obydwu warunków. Blok **SIMTRACE** możemy skasować komenda:

&. kill simtrace-mod

Znajomość pracy systemu przydaje się przy tworzeniu ekonomicznych pytań. Zdania:

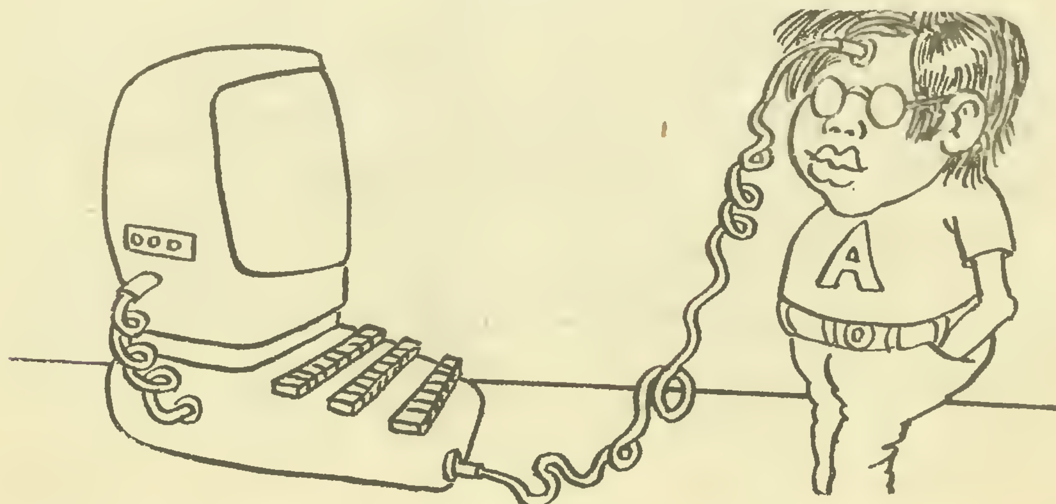
&. which x : Henry father-of x and x male

oraz

&. which x : male and Henry father-of x

dadzą tę samą odpowiedź z tą różnicą, że w pierwszym przypadku Prolog znajdzie wszystkie takie x, że Henry father-of x, a następnie sprawdzi, które spełniają warunek: x male. W drugim przypadku Prolog postąpi wręcz odwrotnie — najpierw znajdzie takie x, dla których wychodzi x male, a następnie sprawdzi, które z nich spełniają warunek: — Henry father-of x. W dużych bazach danych wielokrotnie większych niż nasza RODZINA odpowiedź na pierwsze pytanie zostanie udzielona bardzo szybko w porównaniu z drugim.

ADAM KRAUZE



MAGNETOFON DLA COMMODORE

Współpraca komputera domowego COMMODORE C 64 i VIC-20 z dowolnym magnetofonem kasetowym jest możliwa, jeżeli zastosujemy odpowiedni układ dopasowujący zmieniający sygnał sinusoidalny na sygnał prostokątny o poziomach TTL.

Na początek uprzedzamy naszych czytelników, że **NIE JEST TO KONSTRUKCJA DLA POCZĄTKUJĄCYCH AMATORÓW ELEKTRONIKI**, w pierwszym rządzie ze względu na możliwość uszkodzenia komputera. W związku z tym faktem ograniczymy się wyłącznie do przedstawienia schematów ideowych i nie publikujemy rysunków gotowych płytek. Dla praktyków nie będzie to żadną przeszkodą.

Na rysunku 1 przedstawiono firmowy schemat układu dopasowującego, proponowany przez COMMODORE. Zastosowany tu układ scalony TTL SN74LS14 nie ma niestety odpowiednika produkowanego w Polsce (znajduje się w nim 6 inwerterów Schmitta).

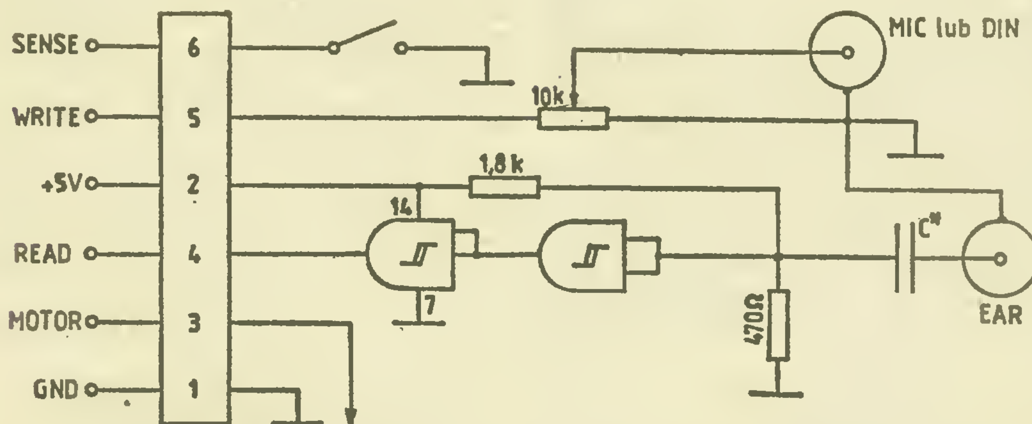
Schemat 2 przedstawia klasyczny przykład przerzutnika Schmitta, proponowany i sprawdzony przez autora. Wykorzystano w nim układ scalony TTL produkowany w Polsce — UCY 74132. Układ ten zawiera 4 bramki z wejściem Schmitta. Do uruchomienia urządzenia z rysunku 2 konieczny jednak będzie oscyloskop. Należy przy jego pomocy tak dobrać kondensator Cx, aby na wyjściu otrzymać jak najbardziej dokładny przebieg prostokątny.

W obydwóch przypadkach potencjometr 10kΩ służy do dopasowania poziomu wyjścia sygnału do czułości wejścia w magnetofonie.

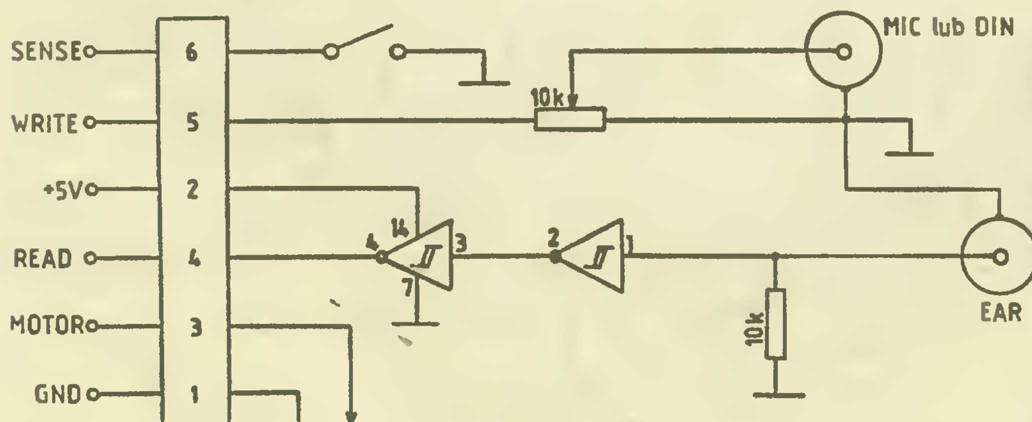
Najwięcej kłopotu sprawi nam zdobycie odpowiedniego złącza krawędziowego do podłączenia łączki do komputera, ale od czegóż jest warszawski „Jarmark Perski"?

Życzę powodzenia w pracy i doradzam ostrożność w trakcie uruchamiania naszego urządzenia.

JAN POZNAŃSKI



Schemat 1



Schemat 2

Znalazłeś się w nieprzebytej, tropikalnej puszczy, pełnej najbardziej fantazyjnych roślin: bardzo pięknych, ale i trudnych do przebycia. Nie ma jednak czasu na podziwianie roślinności: zbyt wiele niebezpieczeństw czyha tu na ciamajdę, któremu zdarza się zagapić.





SABRE WULF

Nie przybyliśmy tu jednak, by podziwiać piękno przyrody, znamy w końcu przyjemniejsze miejsca, a poznamy jeszcze piękniejsze, gdy zdobędziemy skarb ukryty w świątyni w środku puszczy. Aby wejść do niej, musimy jednak zdobyć starożytny amulet, który chroni przed złymi duchami. Jak go odnaleźć, skoro przed wiekami został rozłupany na czworo, a części jego rozrzucono po krańcach puszczy?

Nikt nie wie, gdzie one są. Przed laty miejscowy lud, który traktował fragmenty amuletu jak klejnoty rodzinne, pieczołowicie ukrył je koło swych chat lub w ostępach, na polanach, do których prowadzi tylko jedna droga. W takich miejscach należy ich szukać.

Nie musimy podziwiać roślinności, ale warto poznać występujące w puszczy zioła. Są wśród nich okazy zupełnie wyjątkowe, jak np. czerwony kwiat, który na pewien czas daje nieśmiertelność i pozwala nie obawiać się żadnych drapieżników puszczy. Czar ten zdejmuje niepozorny biały kwiatek przypominający lilię. Niebieska lilia może pomieszać nam zmysły i spowodować, że gdy chcemy iść w lewo — nogi niosą nas w prawo. Lilia zielona — zamienia nas w nieruchomy posąg, na szczęście tylko na chwilę.

Poza roślinami są w dżungli i zwierzęta. Warto je poznać, by wiedzieć, które są niegroźne, a od których lepiej uciekać lub strzelać bez zastanowienia. Do groźnych, ale dających się odpędzić, należą m. in. hipopotamy, nosorożce, dziki. Krokodyle lepiej natychmiast zabić, podobnie jak pająki. Miejscowa ludność również nie odnosi się do naszej wyprawy z pełnym zrozumieniem.

W dżungli możemy natrafić też na różne dziwne przedmioty, jak np. kapelusze, worki itp. Czasem warto je wziąć ze sobą — może uda się je sprzedać?

Z pewnością warto zaopiekować się porzuconym niemowlęciem. Nasze dobre serce bywa w takich wypadkach sowicie wynagrodzone — obdarzeni bywamy dodatkowym życiem, a więc zdolnością do przeżycia jednej, śmiertelnie niebezpiecznej przygody więcej.

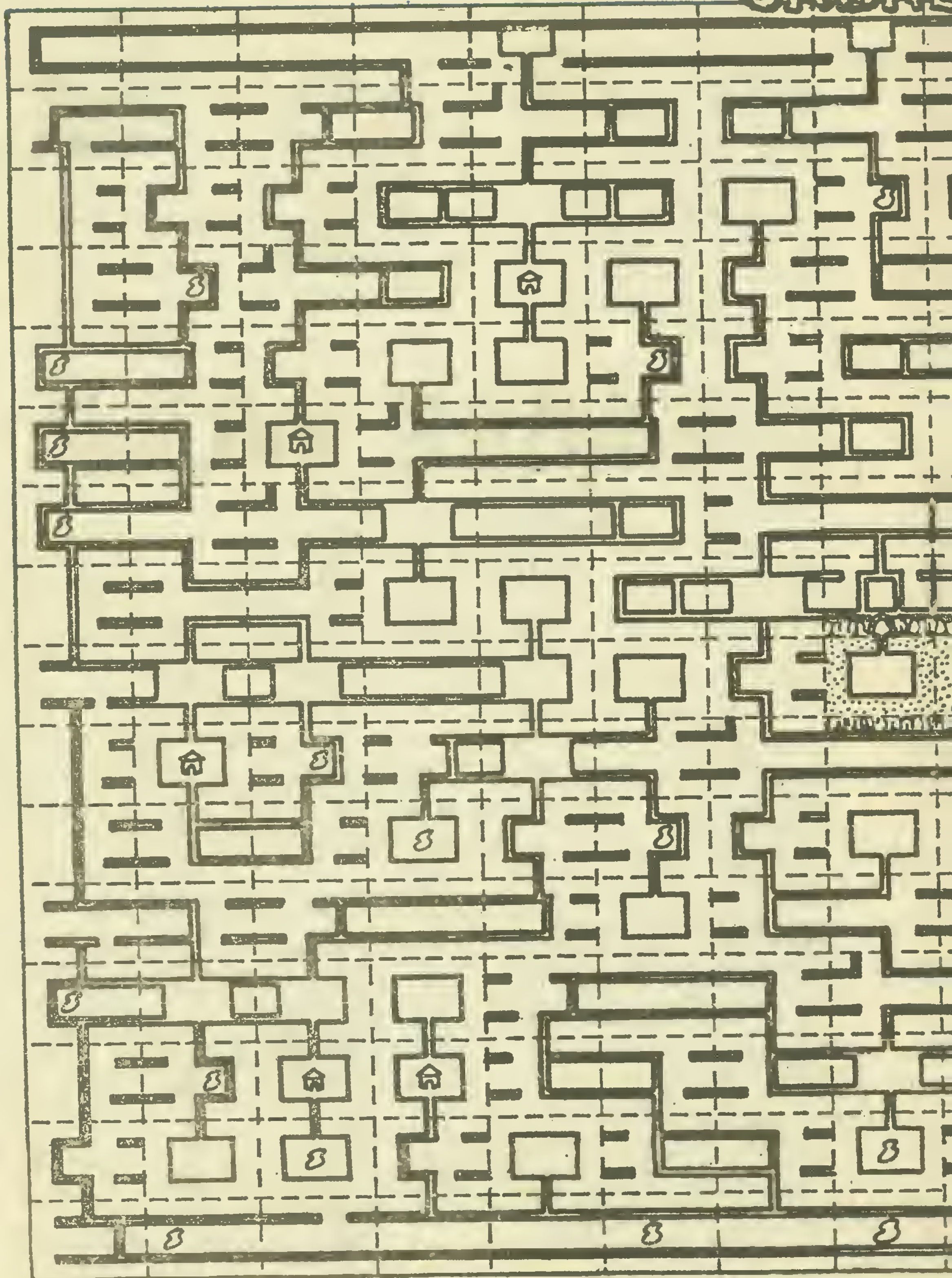
Dlaczego jednak naszej wyprawie daliśmy kryptonim „Sabre Wulf"? Otóż powszechnie jest wiadome, że puszcza ta obok zwykłych niebezpieczeństw kryje jedno szczególne — w jej ostępach, a zwłaszcza w niżej położonej jej części żyje legendarny wilkołak: właśnie Sabre Wulf. Gdy go spotkasz — nie ma dla ciebie ratunku, a zwykle gdy go dostrzeżesz, jest już za późno na ucieczkę czy ukrycie się.

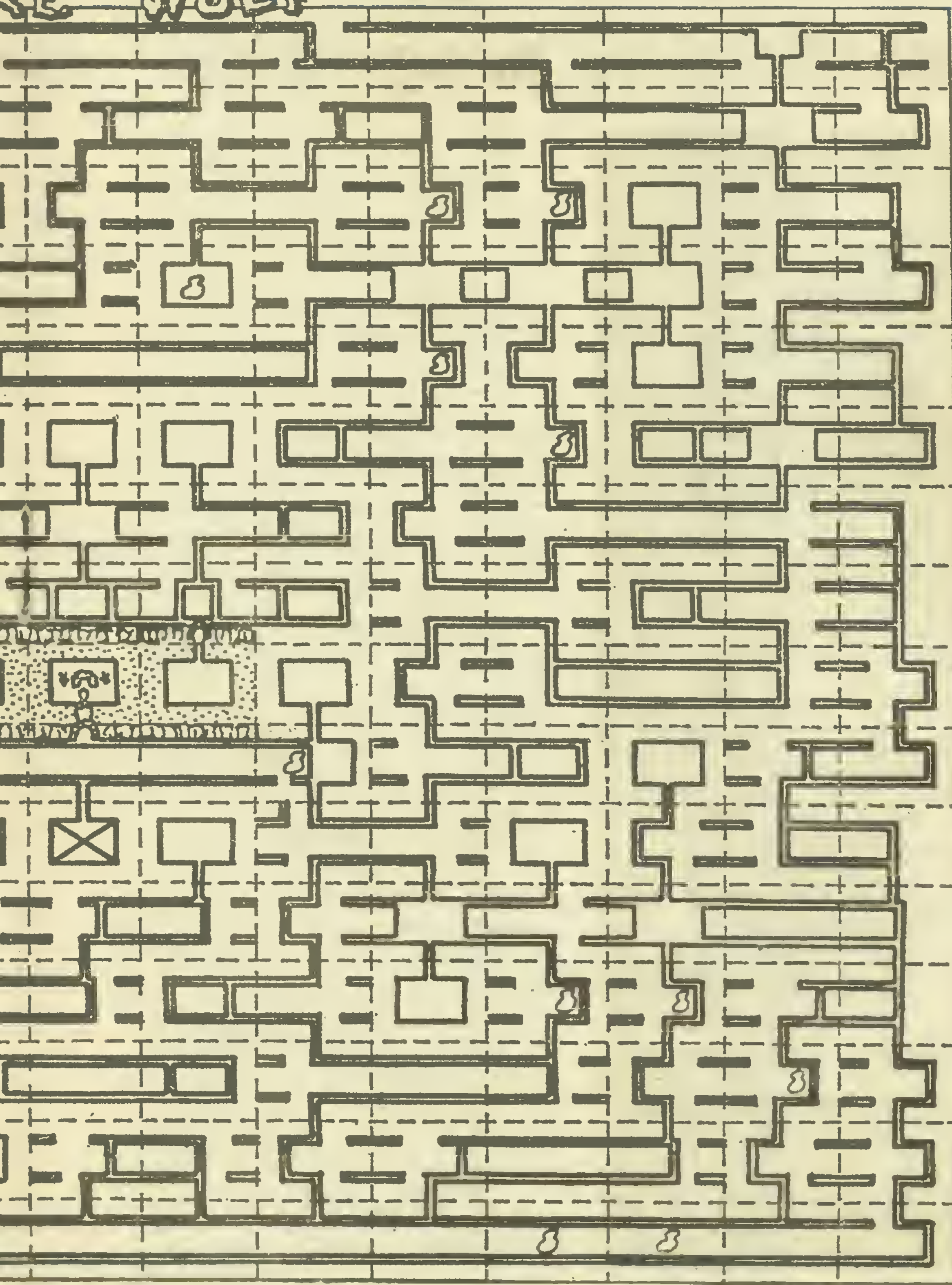
Na szczęście prowadzi on dosyć uregulowany tryb życia i jeśli pozna się jego obyczaje, można uniknąć wchodzenia mu w drogę.

✱ ✱ ✱ Oczywiście nie my pierwsi wybieramy się na tę wyprawę. „Sabre Wulf" była największym przebojem rynku gier na ZX Spectrum w I poł. 1984 r. Łamy „Sinclair User" pełne były listów nastoletnich narkomanów komputerowych, którzy z dumą donosili, że po 48 godzinach nieustannego klepania w klawisze w kilkuosobowych zespołach udało im się zdobyć bajkowy skarb. W następnym numerze kolejny fan chwalił się, że on tej sztuki dokonał w 24 godziny, następny — że w 16 itp. Mapa dżungli, którą publikujemy, pozwoli Wam dokonać tej sztuki w znacznie krótszym czasie. Wielu śmiałków straciło życie, by ją sporządzić. Mapa obejmuje obszar 256 ekranów — 256 różnych miejsc, w których możemy się znaleźć.

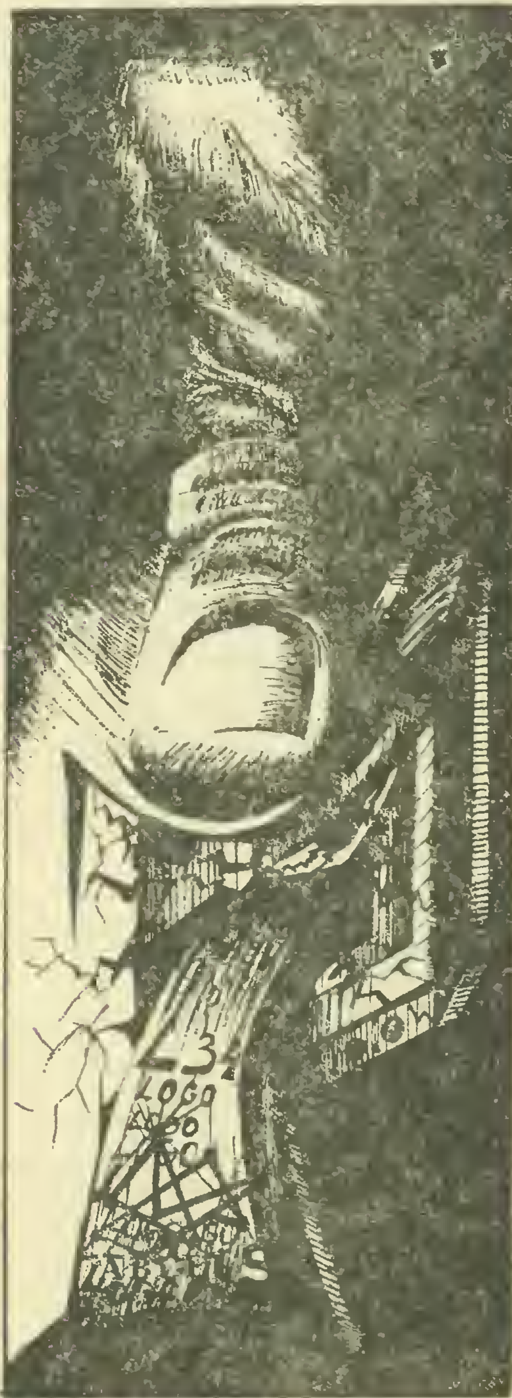
Reszta jest w Twoich rękach, w Twojej zręczności, cierpliwości i inteligencji.

(WM)


SABRE


RE WULF

FLOWCHART czyli sieć działań programu



W październikowym numerze BAJTKA pisaliśmy o algorytmach, o przedstawianiu ich w postaci graficznej i o tym, jak bardzo są one pomocne przy pisaniu programów. Teraz zajmiemy się procesem odwrotnym — analizą i śledzeniem działania gotowego programu.

Okazuje się, że nawet w przypadku stosunkowo prostych i krótkich listingów ich analiza „w pamięci" sprawia sporo kłopotów. I tu znowu przydaje się umiejętność przedstawiania algorytmów w postaci schematów blokowych. Narysowanie takiego schematu na podstawie programu zajmuje co prawda sporo czasu, lecz w gruncie rzeczy nie jest to praca wymagająca zbyt dużego wysiłku intelektualnego — a więc coś w sam raz dla komputera.

Program, którego listę rozkazów przedstawiamy na następnej stronie, został napisany przez Briana Sidesa, na ZX Spectrum. Zawiera on zaledwie 52 linie i z powodzeniem zastępuje nas w tych pracochłonnych czynnościach. Oczywiście dotyczy to programów napisanych w BASIC-u.

Warto zwrócić uwagę na specjalną numerację linii: rozpoczyna się ona od liczby 9900 i jest prowadzona co jeden (bez wolnych numerów). Chodzi o to, by nasz program nie zajął — przypadkiem — tych samych linii, w których zapisany jest program analizowany. Oczywiście, przed przystąpieniem do pracy z każdym programem należy sprawdzić, czy nie zawiera on linii oznaczonych numerami 9900—9951. Jeśli mimo wszystko są takie, należy zmienić ich numerację.

Teraz już możemy czytać nasz program analizujący. Korzystamy w tym celu z instrukcji:

MERGE "FLOWCHART"

i uruchamiamy go przy pomocy polecenia skoku do linii 9900 w trybie bezpośrednim (bez numerowania linii):

GOTO 9900

Przystępując do przeglądania schematu blokowego mamy do wyboru dwie możliwości: możemy przeglądać kolejne jego fragmenty na ekranie monitora, możemy także — pod warunkiem, że dysponujemy drukarką — zapisać go na papierze (komentarze w programie zmieniono na polskie).

Sam schemat blokowy to jednak jeszcze trochę za mało do sprawnej analizy programu, warto — prócz tego — sporządzić sobie tabelkę, w której zapisywać będziemy dane wprowadzane z klawiatury, kolejne wartości zmiennych, polecenia wysyłane na ekran, drukarkę i tor foniczny. Obok przedstawiamy przykład takiej tabelki.

ROMAN POZNAŃSKI

Klawiatura	X_1	X_2	...	X_n	ekran	drukarka	dźwięk

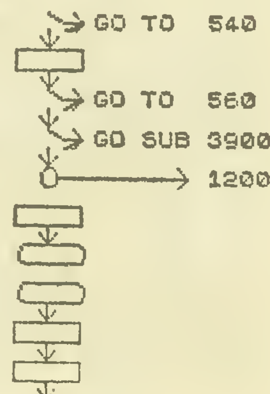
Przykład tabelki niezwykle przydatnej do analizy każdego programu napisanego w BASIC-u.

Sieć działań programu — ZX Spectrum 16/48kB.

```
9900 REM flow-chart
9901 OVER 0: INVERSE 0: BORDER 7
: PAPER 7: INK 0: CLS
9902 LET a=PEEK 23636: LET a=a*2
56: LET a=a+PEEK 23635
9903 LET c=0: LET y=175: LET x=1
30
9904 LET l=PEEK a: LET l=l*256:
LET l=l+PEEK (a+1): LET a=a+4
9905 IF l>9899 THEN GO TO 9943
9906 PRINT l;TAB 5;CHR$ PEEK a:
LET c=c+2: PRINT
9907 PLOT x,y: LET b=PEEK a
9908 IF b=226 OR b=234 OR b=242
OR b=254 THEN GO TO 9924
9909 IF b>227 AND b<231 OR b=232
 OR b=235 OR b=241 OR b=247 OR b
=249 OR b=253 THEN GO TO 9925
9910 IF b=243 OR b=250 THEN GO T
O 9926
9911 IF b=236 OR b=237 THEN GO T
O 9934
9912 DRAW 16,0: DRAW -4,-8: DRAW
 -28,0: DRAW 4,8: DRAW 12,0
9913 LET y=y-8
9914 IF b=226 OR b=236 OR b=237
OR b=254 THEN GO TO 9916
9915 PLOT x,y: DRAW 0,-7: DRAW 4
,4: DRAW -4,-4: DRAW -4,4
9916 LET y=y-8
9917 LET b=PEEK a
9918 IF b=14 THEN LET a=a+4
9919 IF b=13 THEN GO TO 9922
9920 IF b=58 THEN GO TO 9940
9921 LET a=a+1: GO TO 9917
9922 IF c=20 THEN GO TO 9943
9923 LET a=a+1: GO TO 9904
9924 DRAW 12,0: DRAW 4,-8,-PI: D
RAW -24,0: DRAW -4,8,-PI: DRAW 1
2,0: GO TO 9913
9925 DRAW 16,0: DRAW 0,-8: DRAW
-32,0: DRAW 0,8: DRAW 16,0: GO T
O 9913
9926 DRAW 8,-4: DRAW 8,0: DRAW -
4,4: DRAW 4,-4: DRAW -4,-4: DRAW
 4,4: DRAW -8,-4: DRAW 8,4
9927 IF b=243 THEN GO TO 9933
9928 LET a=a+1: LET f=PEEK a
9929 IF f=14 THEN LET a=a+4
9930 IF NOT f=203 THEN GO TO 992
8
9931 LET a=a+1: LET f=PEEK a: PR
INT AT c-2,19;CHR$ f: PRINT
9932 IF f=236 OR f=237 THEN GO T
O 9935
9933 GO TO 9913
9934 CIRCLE x,y-4,4: PLOT x+4,y-
4: DRAW 64,0: DRAW -4,4: DRAW 4,
-4: DRAW -4,-4: PLOT x,y
9935 LET a=a+1: LET f=PEEK a
9936 IF NOT f=14 THEN GO TO 9935
9937 LET a=a+3: LET g=PEEK (a+1)
: LET g=g*256: LET g=g+PEEK (a)
9938 PRINT AT c-2,26;g: PRINT
9939 GO TO 9913
9940 IF c=20 THEN GO TO 9943
9941 LET a=a+1: PRINT "  ▘:";TAB
 5;CHR$ PEEK a: PRINT : LET c=c+
2
9942 GO TO 9907
9943 INVERSE 1: PRINT AT 21,1;"
""p"" -przegladanie, ""w"" -wydr
uk": INVERSE 0
9944 IF INKEY$="p" THEN GO TO 99
47
9945 IF INKEY$="w" THEN GO TO 99
51
9946 GO TO 9944
9947 CLS : LET d=0: LET e=22: LE
T c=0: LET x=130: LET y=175
9948 IF PEEK a=58 THEN GO TO 994
1
9949 IF l>9899 THEN STOP
9950 LET a=a+1: GO TO 9904
9951 PRINT AT 21,0;"▘: COPY : GO T
O 9947
```

rzykładowy wydruk

```
520  IF
530  LET
540  IF
550  IF
   : GO SUB
560  LET
570  RETURN
600  REM
610  LET
620  LET
```



HISTORIA ZŁOTEGO JABŁUSZKA (2)

W poprzednim numerze opublikowaliśmy pierwszą część rozmowy Gregga Wiliamsa i Roba Moora z miesięcznika BYTE ze Stevem Wozniakiem. Była w niej mowa o powstaniu firmy Apple i modelu APPLE I. Dziś — APPLE II.

BYTE: Jak doszło do przekształcenia się Apple I w Apple II?

WOZNIAK: Sprzedawaliśmy Apple I mając przy tym wiele zabawy i zyskując rozgłos. Była to najbardziej niewiarygodna rzecz w naszym życiu. Nadal pracowałem u Hewletta-Packarda, a nocami testowałem płytki Apple, pisałem kolejne programy i projektowałem interface do magnetofonu kasetowego pozwalający wprowadzić BASIC w ciągu kilku minut. Podczas jednego ze spotkań ze Stevem zaprojektowałem małą siedmioukładową przystawkę do Apple I, pozwalającą uzyskiwać kolory na ekranie. Zmontowaliśmy ją i działała!

Miałem wówczas komputer ze swoją własną pamięcią i terminal video z pamięcią obrazu w rejestrach przesuwanych. Zacząłem więc analizować sposoby połączenia obu tych funkcji w jednej grupie układów — aby w jakiś sposób jedna pamięć realizowała obie funkcje. W końcu opracowałem rozwiązanie, w którym drobna część głównej pamięci komputera służyła terminalowi jako pamięć obrazu.

● **Jaka była rozdzielczość koloru?**

● Z początku niska: mógł on wysyłać znaki bądź w trybie tekstowym, bądź jako kolorowe plamki. Aby to osiągnąć, musiałem nieco zmienić prędkości poziomego przebiegu plamki w stosunku do standardow NTSC (National Television Standarts Committee), gdyż NTSC — to oczywiste — nie został ustalony z myślą o wyświetlaniu cyfrowym.

Od czasu zakończenia pracy nad Apple I wszystkie moje myśli dotyczyły Apple II. Uświadomiłem sobie, że większość ludzi nie będzie stać na kolorowy monitor, tak więc nasze urządzenie musi współpracować z domowym telewizorem. Telewizor taki może jednak wyświetlać tylko 40 znaków w jednym wierszu ekranu, tak więc musiałem dostosować organizację ekranu Apple II do możliwości typowego telewizora.

● **A co było przyczyną dziwnego sposobu adresowania grafiki?**

● W układach video miałem liczniki używane do ustalania miejsca wyświetlania plamki w poziomie i w pionie, musiałem odwzorować ich stany w przestrzeni adresowej pamięci obrazu. Jeśli rozważać to czysto teoretycznie, wówczas wydaje się, że nie może być nic prostszego od doskonale liniowego odwzorowania. Wystarczy pomnożyć stan licznika pionowego przez 40 (liczba znaków w linii) i dodać stan licznika poziomego. Zrealizowanie tego wymagałoby jednak trzech dodatkowych układów 7483, a ja przecież szukałem sposobów zmniejszenia liczby układów. Pomyślałem i zauważyłem kilka możliwych trików, które pozwoliły mi osiągnąć cel przy pomocy tylko jednego dodatkowego układu.

● **Słyszeliśmy, że ktoś trzeci pomógł Ci zaprojektować układy wejścia/wyjścia w Apple II. Czy to prawda?**

● Tak. Wiele ówczesnych komputerów miało złącza do przyłączania kart We/Wy, lecz karty te wymagały pęczków układów dekodujących adresy na każdej z nich. Było to rozwiązanie drogie i wymagające dodatkowych kości. Ja lubowałem się w projektach oszczędnych, a równocześnie chciałem mieć osiem przyłączy. Myślałem więc o podzieleniu przestrzeni adresowej między te przyłącza i używaniu tylko jednego dekodera.

W realizacji tej idei bardzo pomógł mi Alan Baum. Miałem już na płytce komputera kilka dekoderów, mogłem więc dekodować co 16 i co 256 adres. Alan zauważył, że każda karta może mieć swój własny 256-bajtowy PROM i że wszystkie te PROM-y mogą dzielić między siebie 2kB przestrzeni adresowej.

● **Jak zdobyliście pieniądze na Apple II?**

● Uzyskaliśmy zamówienie na 1000 płytek kosztujących nas po 250 dol., potrzebowaliśmy więc 250 tys. dol., a nie mieliśmy nic. Plan finansowy pomógł nam stworzyć Mike Markkula. Uwierzył, że rynek mikrokomputerowy jest czymś, co naprawdę ma szanse powstać. Liczył, że komputery domowe będą sterować lodówkami i kuchenkami. Mylił się, nic takiego się nie stało, ale przyłączył się do nas jako trzeci wspólnik.

*

W 1981 r. Wozniak przeżył wypadek lotniczy, po którym przez kilka tygodni miał amnezję, a po wyzdrowieniu postanowił odejść z firmy, która stała się na jego gust za wielka, nie było już w niej miejsca dla samotnego producenta-hobbysty. Pod przybranym nazwiskiem podjął studia w Berkeley, gdzie długo nie został rozpoznany, co oczywiście przyniosło serię zabawnych qui pro quo — w końcu Apple było największym sukcesem w historii biznesu, najpełniejszą realizacją amerykańskiego mitu. O Wozniaku krążyły legendy, z których części kazano mu uczyć się na pamięć. Po roku wrócił do Apple, ale o swej obecnej pracy opowiada bez zachwytu.

tłum. i opr.
Władysław Majewski

OLIMPIADA INFORMATYCZNA

Od ponad trzydziestu lat oswoiliśmy się z pojęciem olimpiady matematycznej, od niewielu mniej — z olimpiadami fizycznymi, astronomicznymi i chemicznymi. Później olimpiady zaczęto organizować nawet z tak trudno wymiernych dziedzin, jak język ojczysty i wiedza o świecie współczesnym. Idea ogólnokrajowego konkursu dla młodzieży szkolnej, mającego wyłonić najzdolniejszych i najwytrwalszych w samodzielnym rozwijaniu swych zainteresowań, zdążyła się więc już utrwalić.

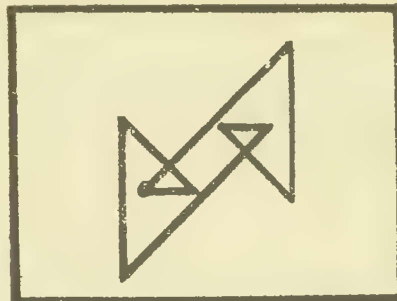
W przyszłym roku w programach niektórych szkół pojawi się nowy przedmiot — elementy informatyki. Już dzisiaj jednak setki tysięcy młodych ludzi (sądząc po powodzeniu BAJTKA jest ich tylu z pewnością) swój wolny czas spędzają przy komputerze, ucząc się współpracy z nim i świadomego jego wykorzystania. Racjonalna wydaje się więc myśl o daniu im szansy rywalizacji, wykazania się inwencją i umiejętnościami, po prostu sprawdzenia się.

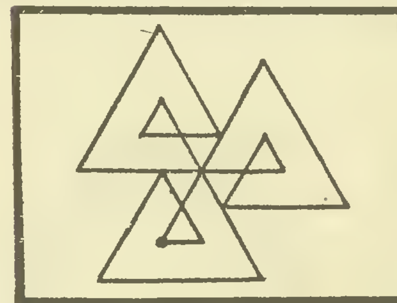
Niestety, w bieżącym roku szkolnym olimpiady takiej nie udało się zorganizować. mimo że najbardziej z urzędu zainteresowane instytucje: Polskie Towarzystwo Informatyczne, Ministerstwo Oświaty i Wychowania, Naczelna Organizacja Techniczna i Towarzystwo Kultury Technicznej w pełni zgadzają się co do potrzeby takiego konkursu. BAJTEK postanowił przejąć inicjatywę i doprowadzić jeszcze w tym roku do spotkania zainteresowanych instytucji, tak by pierwsza olimpiada mogła odbyć się już w roku szkolnym 1986/87. O jego wynikach i podjętych decyzjach poinformujemy już wkrótce!

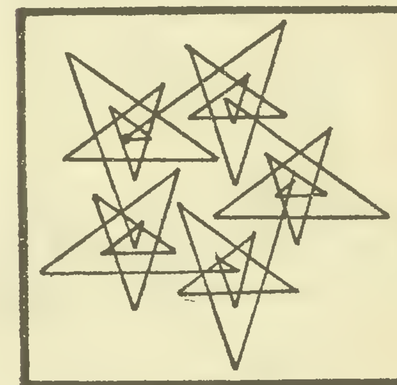
Tymczasem proponujemy Wam zapoznanie się z zasadami konkursowymi podobnych olimpiad odbywających się w innych krajach. Jeśli postanowicie spróbować swych sił i rozwiązać publikowane dziś zadania — chętnie zapoznamy się z wynikami Waszych dociekań. Najlepsze z nich opublikujemy.

MOSKIEWSKA ZAOCZNA OLIMPIADA PROGRAMOWANIA

Znakomity popularnonaukowy matematyczno-fizyczny miesięcznik Akademii Nauk ZSRR „Kwant”, (odpowiednik naszej DELTY) w numerze 6/85 opublikował zadania finału szóstej już olimpiady programowania dla uczniów moskiewskich szkół średnich, zorganizowanej przez Dydaktyczno-Produkcyjne Centrum Techniki Obliczeniowej okręgu Oktiabrskiego Moskwy oraz Instytut Elektronicznych Maszyn Sterujących i Problemów Informatyki Akademii Nauk Związku Radzieckiego. W zawodach tych uczestniczyło około 200 uczniów szkół moskiewskich. „Kwant” publikując te zadania ogłosił równocześnie dla wszystkich czytelników spoza Moskwy Zaoczną Olimpiadę Programowania.







Rys. 1

Ocenę rozwiązań obniżają pomyłki oraz algorytmy z nadmierną liczbą działań lub blokami, bez których można się obejść. Dopuszczalne są rozwiązania w dowolnym języku programowania. Rozwiązanie należy poprzedzić słownym objaśnieniem algorytmu, a ponadto w tekście programu powinny znajdować się niezbędne komentarze. Programy powinny być uporządkowane zgodnie z regułami programowania strukturalnego.

A oto zadania:

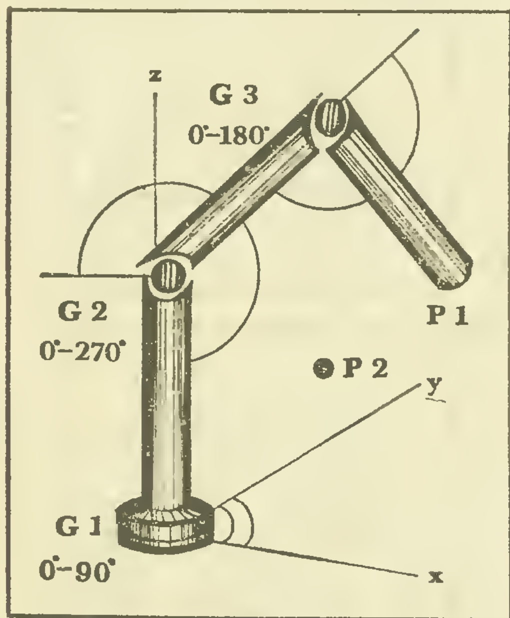
1. **Rozkład na składniki:** Wypisać wszystkie sposoby przedstawienia liczby naturalnej n w postaci sumy liczb naturalnych. Permutacje składników nie są traktowane jako sposoby różne.

2. **Równe elementy**. Dana jest macierz liczb P (1:m, 1:n). Każdy wiersz macierzy uporząd-

kowany jest rosnąco. Znaleźć i wypisać liczbę, która występuje we wszystkich wierszach lub słowo NIE, jeśli liczba taka nie istnieje.

3. **Nierozkładalna liczba.** Dany jest podzbiór liczb naturalnych M (1:n). Znaleźć i podać najmniejszą liczbę naturalną nie będącą sumą elementów zbioru. Suma ta w szczególnym przypadku może się składać tylko z jednego składnika, każdy element zbioru może się w niej znaleźć tylko raz.

4. **Czworościany.** Na ścianach dwóch jednakowych czworościanów prawidłowych wypisano odpowiednio liczby M1, M2, M3, M4 na jednym oraz N1, N2, N3, N4 na drugim (w podanej kolejności). Czy można nałożyć na siebie czworościany w ten sposób, by na pokrywających się ścianach znalazły się jednakowe liczby?



Rys. 2

5. **Najczęściej występująca liczba.** W zbiorze liczb M (1:n) znaleźć liczbę występującą najczęściej. Jeśli ich jest kilka — podać jedną z nich.

6. **Systemy liczbowe.** Zbiór liczb całkowitych M (1:9) zawiera cyfry przedstawienia pewnej liczby naturalnej w systemie liczbowym o podstawie i (tzn. M (1) oznacza liczbę jedności w tym przedstawieniu tej liczby itd.). Należy przedstawić tę liczbę w systemie liczbowym o podstawie j, gdzie liczby i i j nie są większe od 10.

7. **Przekątna boczna.** Znaleźć sumę elementów A (i, j) macierzy A (1:m, 1:n), mających stałą różnicę indeksu i — j=k. Liczba k jest całkowita, ale niekoniecznie dodatnia.

Wyjaśnienia: Macierz P (1:m, 1:n) oznacza macierz Pij dla i=1, 2, ..., m oraz j=1 ...,n, natomiast zapis P (i, j) — element Pij. Przez formułowania „dana jest macierz P (1:m, 1:n)" należy rozumieć, że dane są liczby m, n i wartości elementów macierzy.

W rozwiązaniach dane wartości należy wprowadzać z zewnątrz i drukować je. Pożądane jest posługiwanie się programami (w ALGOLU — procedura, w FORTRANIE — subroutine itd.). W tych językach, w których wymiary macierzy należy zadawać liczbowo, np. w FORTRANIE, można je określić dowolnie.

„Dobre rozwiązania" dla dużych liczb danych powinny składać się z dokładnością do stałego czynnika z następującej liczby działań: m n (zadanie 2), n^2 lub n log n (zadania 3 i 5), min. m, n (zadanie 7).

IV FEDERALNY KONKURS INFORMATYCZNY 1985

Federalne Ministerstwo Nauki i Oświaty RFN ogłosiło (pod osobistym patronatem pani minister Dorothee Wilms) IV Federalny Konkurs Informatyczny 1985. Bezpośrednimi organizatorami konkursu są Federalne Towarzystwo Matematyczne i Przetwarzania Danych oraz Federalne Towarzystwo Informatyczne. To ostatnie liczy sobie 8500 członków — podaję dla zilustrowania, że w krajach zachodnich informatycy nie są wcale tak liczni — różnica w efektach wynika z intensywności pracy i uzbrojenia technicznego informatyków.

Pierwsza seria zadań konkursowych ogłoszona została 1 września 1985 r., a rozwiązania należało nadsyłać do 15 listopada. Druga runda — dla tych, którzy poprawnie rozwiązali zadania pierwszej serii — przewidziana jest w okresie luty—kwiecień 1986 r., a finały na specjalnym spotkaniu uczestników w sierpniu 1986 r.

Organizatorzy oczekiwali, że rozwiązania zadań zawierać będą: ideę rozwiązania, dokumentację programu, protokół testowego przebiegu programu i oczywiście listę rozkazów.

1. **Jak to zaprogramowano?** Rysunki 1 przedstawiają wynik trzykrotnego wykonania programu z dwoma parametrami: KĄT i LICZBA, przy podanych wartościach tych parametrów. Napisz program dający takie właśnie rezultaty oraz wykonaj go dla KĄTA równego 70° i LICZBY równej 9.

2. **Jam Mata Hari, Bond i Kloss.** Rozszyfrować tekst: CD RDBAOE OZSOYFADWZNE TAURND DRSOYFADWZC, wiedząc, że jest to zdanie w języku polskim, w którym pięć najczęściej występujących liter przetasowano (poddane permutacji), a pozostałe znajdują się bez zmian na swoich właściwych pozycjach.

Przykład: tytuł zadania po zaszyfrowaniu brzmi: JMI IMTM HMRA BSND A KLSOO, po podstawieniu M za A, I za M oraz A za I i zamienieniu miejscami S i O.

Napisany przez Was program powinien odczytywać dowolny tekst zaszyfrowany według tej zasady, przy czym rozpoznawanie, która z możliwych permutacji prowadzi do sensownego odczytania, można pozostawić użytkownikowi.

3. **Sterowanie ramieniem robota.** Należy przesunąć koniec ramienia robota z punktu P_1 do punktu P_2, przy czym punkty te nie muszą leżeć na jednej płaszczyźnie. Ramię robota składa się z trzech części o długości 180 cm każda (niezła łapa, co?) i trzech przegubów. Pierwszy z nich (G1) pozwala ramieniu obracać się wokół osi pionowej (od 0 do 90°), natomiast drugi (G2) i trzeci (G3) wokół osi poziomej — odpowiednio od 0 do 270° i od 0 do 180° (patrz rysunek 2).

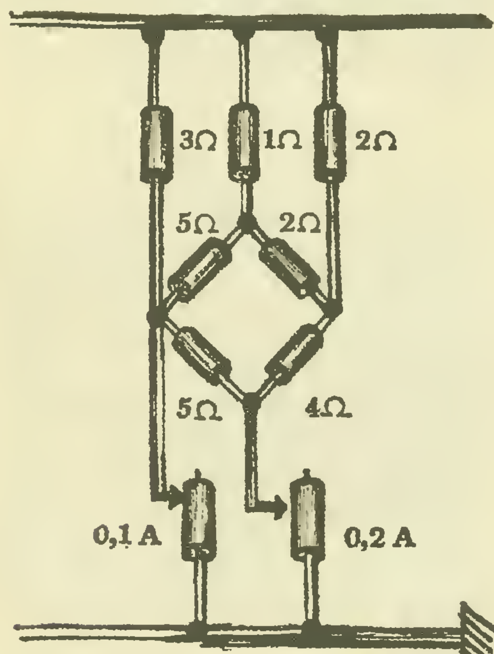
Sterowanie ramieniem robota odbywa się przy pomocy rozkazu TWIST (i, j, k), gdzie

i, j, k mogą przyjmować wartości -1, 0 i 1°. Dla przykładu rozkaz TWIST (0, -1, 1) oznacza pozostawienie przegubu G1 bez zmian, podczas gdy pozostałe kąty będą odpowiednio zmniejszone i zwiększone o 1° (oczywiście dopóki nie zostaną osiągnięte kąty graniczne — w takim wypadku odpowiedni rozkaz jest ignorowany).

a. Napiszcie program, który przesuwa ramię robota z punktu x_1, y_1, z_1 do punktu x_2, y_2, z_2 lub możliwie blisko tego drugiego punktu (położenie punktu określone jest we współrzędnych kartezjańskich). Przetestuj swój program m.in. dla przypadku P_1 = (100 cm, 0, 0) i P_2 = (100 cm, 100 cm, 0)...

b. Postaraj się zapisać swój program w taki sposób, by czubek ramienia robota poruszał się w miarę możliwości po prostej (to są dwa różne zadania — w pierwszym dopuszczamy ruch inną drogą, jeśli pozwoliłoby to uprościć program).

4. **Analiza kursów giełdowych.** Treść tego zadania tak dalece nie ma odniesień w naszej rzeczywistości, że pomijamy je w BAJTKU



Rys. 3

5. **Symulacja sieci oporników.** Bierzemy pod uwagę sieci spełniające następujące warunki:

a. Sieć jest płaska (tj. nie występują „dwupoziomowe" skrzyżowania przewodów.

b. Połączenia z ziemią poprowadzone są poprzez potencjometry ustawione w taki sposób, by danymi odgałęzieniami płynął z góry zadany prąd.

c. Dopuszczalne są różne napięcia całkowite i różne wartości oporów.

d. W węzłach mogą się schodzić więcej niż trzy przewody.

Podajcie program, który będzie obliczał rozkład prądów i napięć w takiej sieci i wyświetlał wyniki w postaci przejrzystej tablicy. Wypróbujcie działanie tego programu m.in. dla sieci podanej na rysunku (3). **Opr. W.M.**

W następnym numerze m.in.:

- Sekrety nagłówków
- LOGO po polsku
- Lądowanie na księżycu
- Sinclair QL
- Komputery w Chinach
- Sygnalizator do Dattasette
- ANT ATTACK
- Sklep z komputerami

Wszystkich zainteresowanych podnoszeniem kultury informatycznej w naszym kraju, osoby posiadające doświadczenie w użytkowaniu mikrokomputerów, wszystkich, którzy chcą z BAJTKIEM współpracować lub choćby wymienić uwagi, prosimy o kontakt z redakcją. Adres do korespondencji:

Redakcja „Sztandaru Młodych"
00-687 Warszawa
ul. Wspólna 61
BAJTEK

Telefony:
29-51-06 (Waldemar Siwiński),
28-52-71 wew. 259
(Roman Poznański).

KOMPUTEROWA FURIA

Korespondencja z Bułgarii

„Komputer przestaje być nowością. Nowością staje się odkrywanie, jakim jeszcze celom może służyć” — z opinią Nikołaja Tomowa, naczelnego inżyniera sofijskiej „Elektroniki” nie sposób się nie zgodzić. Zawarta jest w niej fascynacja Bułgarów wyścigiem za światową czołówką, fascynacja elektroniką i jej wytworami.

A zaczął się ów wyścig tak przecież niedawno... Jeszcze w końcówce lat sześćdziesiątych i na progu siedemdziesiątych nasi południowi pobratymcy raczkowali w tej dziedzinie. Dzisiaj eksportują swoje „maszynki” do 60 krajów świata (w ciągu pięciu lat eksport wzrósł trzykrotnie). Trzydzieści procent rocznej produkcji tutejszego przemysłu elektronicznego stanowią zupełnie nowe urządzenia. Ten błyskawiczny awans spowodował, iż RWPG powierzyła Bułgarom rolę koordynatora w rozwijaniu i wdrażaniu do produkcji robotów przemysłowych. Bułgarzy chwalą się — i słusznie — tą produkcją. Najbardziej jednak fascynują



IZOT 1036 s — 16-bitowy profesjonalny mikrokomputer, produkowany w Bułgarii

KOMPUTERY OSOBISTE

Prace nad pierwszym urządzeniem tego typu rozpoczęto tutaj w 1978 roku. Już po dwóch latach **IMKO-1** (indywidualny komputer) był gotów. Rok później jego udoskonalona wersja **IMKO-2** stała się podstawą do masowej produkcji. W 1982 roku zakłady w **Prawec** wypuściły pierwszą serię bułgarskich „komputerów dla każdego”. Produkują je przede wszystkim zakłady w **Prawec**. W ubiegłym roku dały one na rynek 5 tysięcy egzemplarzy komputerów osobistych, tegoroczna produkcja ma się zamknąć w 12 tysiącach.

Robione jeszcze ciągle „PRAWEC-2” mają licznych następców. Trwają właśnie prace nad komputerem 8-bitowym z 48-kilobitową pamięcią. Mają to być urządzenia „wielojęzyczne”, którym nieobcy będzie ani **BASIC**, ani **PASCAL**, ani **FORTRAN**. Dyrektor zakładów — **Płamen Waczkow** zdradził ostatnio prasie tajemnicę (takie tajemnice zdradza się tu z dumą i chętnie), iż w przyszłym roku rusza produkcja komputera 16-bitowego z pamięcią 64 kB. — W ten sposób dogonimy świat i wiodące firmy — oznajmił.

DOMOWE, UNIWERSALNE, PROFESJONALNE

Takie właśnie komputery robi aktualnie Bułgaria — od 8 do 16 bitów, w zależności od klasy.



Prawec-16 — najmłodsze dziecko w rodzinie Prawców

Pełną parą idzie już produkcja 16-bitowych komputerów **„IMKO-4”** i **„MID-16”** (we współpracy z amerykańską firmą **IBM**), rusza produkcja serii **„Prawec-16”**. Szlagierem sezonu są profesjonalne **„IZOT-y”: 1036s, 1037s i 1039s** z 16-bitowymi procesorami.

90 proc. produkcji sofijskiej **„Elektroniki”** idzie na eksport. Zakład, tylko w tym roku, uruchomił wytwarzanie 24 nowych urządzeń. Nie sposób wymienić wszystkich typów.

A CO DLA LUDZI?

Podochoceni decyzją Biura Politycznego KC BPK obywatele (jest to decyzja z jesieni ubiegłego roku, mówiąca o powszechnej edukacji komputerowej) ślą listy do różnych redakcji z pytaniem, kiedy komputer będzie można kupić w sklepie? Producenci i handlowcy twierdzą, że



Tak wygląda bułgarski odpowiednik BAJTKA



Klucz do komputera czyli wprowadzenie do BASIC'a — podręcznik dla najmłodszych

już wkrótce. Póki co, absolutne pierwszeństwo mają jednak szkoły i kluby komputerowe, nie mówiąc rzecz jasna o przemyśle, rolnictwie i administracji, które w całości prawie zostały skomputeryzowane. Robi się już (w Prawec) komputery domowe, wprawdzie bez monitorów, ale za to z możliwością współpracy z telewizorem. Przypominają maszyny do pisania, a zastąpić mają domowników np. w myśleniu o opłatach, o tym co i od jak dawna znajduje się w lodówce. Zresztą od pomysłowości i umiejętności ich przyszłych właścicieli będzie zależało, w czym komputer domowy pomoże gospodarzom. Na razie koszt takiego cacka jest prawie abstrakcyjny — około 5 tysięcy lewów.

Wspomniany już program Biura Politycznego, zmierzający do powszechnej komputerowej edukacji, realizowany jest tu

Z ŻELAZNĄ KONSEKWENCJĄ

W kompleksowym programie powszechnej komputerowej edukacji mówi się, iż do 1990 roku trzeba przeszkolić 30 tysięcy osób. Kontakt z komputerem muszą mieć więc wszyscy młodzi — uczniowie szkół średnich, studenci, na razie hobbistycznie — uczniowie podstawówek i eksperymentalnie... przedszkolaki. Co robi się, by ta dyrektywa była realna?

W szkołach średnich Warny pracuje 51 komputerów osobistych **„Prawec-82"**. Ministerstwo Oświaty do końca roku „obiecało" dostarczyć (na razie przydzieliło) kolejnych 160. Chodzi o to, aby mikrokomputerami obdzielić nie tylko uczniów z wielkiego miasta, ale i z małych miejscowości. W okręgu wraceńskim z komputerami mają kontakt wszyscy uczniowie dziewiątych klas. Chociaż, jak to zwykle bywa, zdarzają się kłopoty. **Georgi Nikołow,** dyrektor szkoły we **Wracy,** skarży się np., że ma u siebie osiem egzemplarzy **„Prawec-82"**, ale dwa są zepsute, a na pozostałych sześciu pracuje dziewięciuset uczniów.

Sofijskie szkoły wyposażono w 650 komputerów osobistych (także **„Prawec-82"**). Największą ich bazą dysponuje Stołeczne Technikum Elektroniczne — 42 sztuki **„Prawec"**. Obsługują jednak nie tylko uczniów szkoły noszącej imię Lenina. Komputerowa baza jest jednocześnie własnością klubu, z którego korzysta młodzież z całej leninowskiej dzielnicy.

W Bułgarii nie ma zwyczaju licytowania się

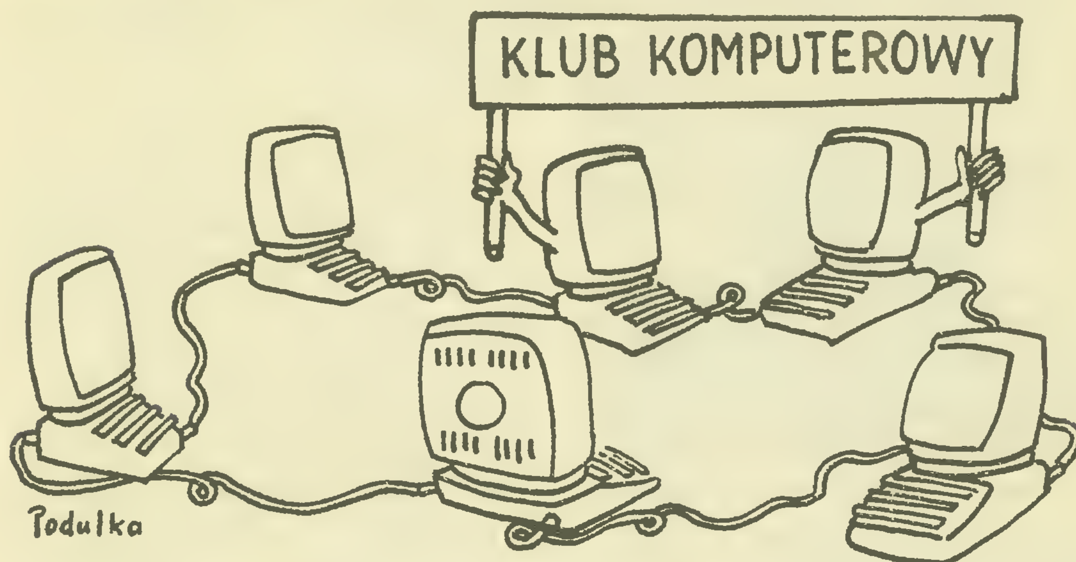
KTO BYŁ PIERWSZY

ale faktem jest, iż to właśnie **Dymitrowski Komsomoł** rzucił hasło powszechnej komputerowej edukacji, wyprzedzając o rok decyzję **Biura Politycznego BPK.** Organizacja komsomolska nie jest i nie była w stanie udźwignąć takiego zadania samodzielnie. Mimo to większość funkcjonujących klubów komputerowych (jest ich ponad 100) pracuje pod jej patronatem. Są wprawdzie otwarte dla wszystkich, ale pierwszeństwo mają zorganizowane grupy.

Dzisiaj nie ma już chyba w Bułgarii komsomolskiego domu młodzieży, w którym nie pracowałby na pełnych obrotach elektroniczny mądrala. Organizacja w dużych nakładach i na pięknym papierze wydaje dla entuzjastów pismo **„Komputer dla Ciebie"**, ukazał się piękny, kolorowy podręcznik **„Klucz do komputera — lekcja języka BASIC"** tak przystępny, że polecić go można nawet absolutnym dyletantom (choć inżynierom też się podoba!).

Rozmach, z jakim Bułgaria wkroczyła na światowe rynki ze swą elektroniczną ofertą, z jakim kształci się komputerowe pokolenie — naprawdę imponuje!

MAGDA RULSKA



Pierwszym w Polsce klubem mikrokomputerowym był ABAKUS. Powstał on na początku 1983 roku w Warszawie, a jego założycielem był Leszek Wilk, obecny prezes klubu. Leszek przekazał na ten cel swój prywatny sprzęt, zresztą do dzisiaj klub pracuje prawie wyłącznie na sprzęcie ofiarowanym przez niego.

Początkowo **ABAKUS** działał przy Klubie Osiedlowym Wilanów, później przeniósł się do Młodzieżowego Centrum Gier i Rozrywek ZSMP w Alejach Jerozolimskich, a obecnie mieści się w Młodzieżowym Domu Kultury przy ul. Brożka 18.

Od samego początku ABAKUS miał charakter klubu otwartego dla wszystkich chętnych. W okresie swego istnienia zorganizował około dwustu pokazów, na które wstęp (zwykle bezpłatny) miały wszystkie osoby zainteresowane techniką komputerową. Organizowane były również kursy programowania i obsługi mikrokomputerów, przeznaczone dla różnych grup wiekowych i zróżnicowanego poziomu wiedzy. Trzykrotnie ogłaszany był konkurs na grę edukacyjną.

Jednym z najważniejszych zadań, jakie postawili przed sobą Leszek Wilk i jego klubowi koledzy, jest popularyzowanie informatyki wśród najmłodszych, a w praktyce wśród dzieci z początkowych klas szkoły podstawowej. Dla osiągnięcia tego celu konieczne było wyjście poza zwykłą działalność klubową. Na początku tego roku ABAKUS brał udział w zorganizowaniu „komputerowego" zimowiska dla dzieci. Eksperyment zakończył się powodzeniem. Podczas wakacji już blisko 500 dzieci, przebywających na koloniach w Tleniu, w Borach Tucholskich, miało okazję zawrzeć znajomość z komputerami.

Członkiem klubu może zostać tylko osoba dorosła, która ukończyła 18 lat. W imieniu dzieci deklarację członkowską wypełniają rodzice. Ze względu na dużą liczbę chętnych i ograniczone możliwości klubu wprowadzono dodatkowy warunek — przyjęcie każdego nowego członka musi być akceptowane przez Radę Klubu. Trzeba więc wykazać się aktywnością, najpierw trochę popracować.

ABAKUS

Warszawa, ul. Brożka 18
tel. 27-87-73 wew. 3, 36-27-49

Związek Harcerstwa Polskiego interesował się sprawami informatyki już od początku lat 80-ych. Mikrokomputery nieraz pojawiały się na harcerskich obozach i zimowiskach.

W październiku 1984 roku w Ośrodku Harcerskim w Chorzowie otwarty został pierwszy harcerski klub mikrokomputerowy

INFORMIK

Inicjatorami powstania klubu byli uczniowie, studenci, młodzi inżynierowie. Nie są przeznaczone głównie dla młodzieży szkolnej. Członkiem INFORMIK-u może zostać każdy, niezależnie od miejsca zamieszkania, przynależności organizacyjnej, profilu szkoły. Okazało się, że amatorzy małej informatyki rekrutują się nie tylko spośród uczniów szkół średnich, lecz także i podstawowych.

Początkowo klub posiadał 5 mikrokomputerów MERITUM I produkcji polskiej, obecnie jest ich już 15 (w tym 10 z rozszerzoną pamięcią), są też cztery ZX SPECTRUM.

Klub posiada bogatą bibliotekę literatury i czasopism — głównie w języku angielskim (co podobno bardzo pozytywnie wpływa na postępy w nauce tego języka wśród członków) a także własne tłumaczenia zagranicznych podręczników i instrukcji.

Obecnie **INFORMIK** posiada około 700 członków z całego kraju. Oprócz tego drzwi klubu są stale otwarte dla tych, którzy chcieliby zobaczyć, co to jest mikrokomputer, a tych jest znacznie więcej.

INFORMIK
Ośrodek Harcerski Katowickiej Chorągwi ZHP
Wojewódzki Park Kultury i Wypoczynku w Chorzowie

Związek Socjalistycznej Młodzieży Polskiej postawił sobie ambitne zadanie stworzenia sieci ogólnodostępnych klubów w miejscach pracy, nauki i zamieszkania młodzieży w całej Polsce. Mają być one budowane w powiązaniu z programem i strukturami Turnieju Młodych Mistrzów Techniki i nosić nazwę

KLUBY KOMPUTEROWE MŁODYCH MISTRZÓW TECHNIKI „POLIN"

Pierwszym klubem, który powstał w ramach ZSMP, jest katowicki **„NEW BIT"**. Obecnie działa już kilkanaście następnych, m.in. „Signum" przy inowrocławskim „Agromecie" — „Inofama".

Struktura sieci klubów komputerowych zbudowana jest na trzech poziomach: podstawowym, wojewódzkim oraz centralnym. Kluby szczebla podstawowego powstają przy szkołach, uczelniach, w zakładach pracy i domach kultury, na tomiast kluby wojewódzkie tworzone są przy Zarządach Wojewódzkich ZSMP.

Organizatorzy liczą, że w niedługim czasie uda się stworzyć tak dużą liczbę tych placówek, iż informatyka stanie się dostępna dla każdego, kto będzie nią zainteresowany — niezależnie od tego, czy mieszka w dużym mieście, czy na wsi.

Nikt dzisiaj nie wie, ile jest w Polsce klubów komputerowych i gdzie się one znajdują. A BAJTEK powinien wiedzieć — domagają się tego nasi czytelnicy.

Prosimy więc o wszelkie informacje o formalnych i nieformalnych grupach fanów informatyki.

Piszcie o tym, co Was najbardziej interesuje w pracy klubowej. Opublikujemy Wasze listy i podamy adresy.

Poznajmy się!

Centralne Biuro TMMT
Warszawa, ul. Smolna 40
tel. 26-42-67

KLUB MIKROKOMPUTEROWY „APPLE"

powstał dnia 8.XII.1984 r. w Warszawie, działa przy Klubie Użytkowników Mikroprocesorów. **Apple** uczestniczy w programowaniu nowych koncepcji programów edukacyjnych związanych m.in. z językiem **LOGO, MuSIMP, MuLISP** oraz zestawem ciekawych problemów matematyczno-fizycznych. Klub dysponuje własnym bankiem programów. Pełne oprogramowanie wraz z dokumentacją jest dostępne członkom bezpłatnie, z zastrzeżeniem, by materiały te nie były wykorzystywane w celach handlowych. Klub posiada kontakty z klubami i przedstawicielami **Apple'a** w Europie Zachodniej, Stanach Zjednoczonych oraz Bułgarskiej Republice Ludowej. W najbliższej przyszłości planuje nawiązanie ściślejszej współpracy z Bułgarią i sprowadzenie stamtąd większej ilości PRAVEC'ów (bułgarskich kopii Apple'a) dla Klubów Apple'a, Kwant i innych. Przed dwoma miesiącami doszło do połączenia klubu Apple z siecią klubu Kwant.

Klub mikrokomputerowy Apple-Kwant
Zakład Biofizyki Instytutu Geologii
Warszawa, ul. Żwirki i Wigury 93

Najnowszym fragmentem historii mikrokomputerowego ruchu klubowego jest

HARCERSKIE CENTRUM KOMPUTEROWE PUŁAWY

którego otwarcie nastąpiło 15 września. Promotorami i w części fundatorami wyposażenia były: **Przedsiębiorstwo Polonijno-Zagraniczne „Polbrit", Gdańskie Zakłady Elektroniczne UNITRA-UNIMOR** oraz **Urząd Miejski w Puławach.**

Centrum mieści się w **Puławskim Ośrodku Kultury „Dom Chemika"** i wyposażony jest w 20 komputerów ZX SPECTRUM. Może z niego korzystać młodzież szkolna i harcerska. W dni wolne od pracy, soboty i niedziele sprzęt udostępniany jest młodzieży w sposób indywidualny, a także — za symboliczną opłatą — innym użytkownikom i instytucjom.

Harcerskie Centrum Komputerowe
Puławy
Puławski Ośrodek Kultury „Dom Chemika"

Projekt zawiązania organizacji zrzeszającej kluby mikrokomputerowe narodził się podczas **„Dni Kultury Mikroinformatycznej"** zorganizowanych na przełomie maja i czerwca br. przez warszawski klub **„ABAKUS"**. Powołana wówczas grupa inicjatywna zobowiązana została do zorganizowania spotkania założycielskiego, do którego doszło 29 czerwca. Tak powstała

POLSKA FEDERACJA KLUBÓW MIKROKOMPUTEROWYCH

Jednym z najbardziej dyskutowanych tematów był projekt statutu przygotowany przez grupę inicjatywną. Powołana została również **Rada Tymczasowa**. Ustalono, że w listopadzie odbędzie się zjazd klubów z całej Polski, który ponownie dokona wyboru **Rady Federacji** i sprecyzuje program działania.

R. P.

Czy mikrokomputery zostaną wprowadzone do szkół i liceów ogólnokształcących, a jeżeli tak, kto będzie mógł z nich korzystać?

Czy istnieją szanse zakupienia w Polsce mikrokomputerów (jeśli tak, to jakich i jaka jest ich cena?)

Stanisław Klęk
ul. Kościuszki 15a 45
33-100 Tarnów



Na pytania czytelników odpowiada Marcin Waligórski, lat 20, student drugiego roku Informatyki Uniwersytetu Warszawskiego.

FOT: Michał Kułakowski

Tak, mikrokomputery będą, w miarę istniejących możliwości, stawać się również wyposażeniem szkół. W pierwszej kolejności będą to szkoły średnie, dopiero w dalszej — podstawowe. Czynione są starania, aby sprzęt mikrokomputerowy, który będzie w szkołach użytkowany, był sprzętem wyższej klasy niż sporadycznie dziś spotykane w szkołach komputery typu Meritum czy ZX Spectrum 16 KB, oraz by był wyposażony w dostateczną ilość urządzeń peryferyjnych, takich jak drukarki czy stacje dyskietek. Wymaga to oczywiście bardzo znacznych nakładów finansowych i organizacyjnych. Proces informatyzacji oświaty będzie zatem



przebiegać stopniowo i trudno jest się spodziewać natychmiastowych sukcesów w tej dziedzinie.

Mikrokomputery szkolne będą użytkowane przez uczniów i nauczycieli szkół średnich nie tylko w ramach przedmiotu „Elementy informatyki", lecz również w ramach kół zainteresowań i jako pomoc dydaktyczna w nauce innych przedmiotów.

Odpowiedź na pytanie drugie: rzecz jasna, szanse takie są. Sprzedaż mikrokomputerów Spectrum 48 kB i Spectrum Plus prowadzi kilka firm polonijnych. Orientacyjne ceny wynoszą 200—220 tys. zł za Spectrum oraz ok. 300 tys. zł za Spectrum Plus. Z punktu widzenia nabywcy jest jednak bardziej opłacalne nabycie tego komputera w handlu komisowym lub na „perskim". O cenach na „perskim" dowiedzieć się można z 2 numeru BAJTKA, ja natomiast podaję kilka przykładowych cen komisowych:

ZX Spectrum Plus — 150 tys.
ZX Spectrum 48 kB — 100—120 tys.
Commodore 16 — 100 tys.
Atari 600 (16 kB) — 100 tys.
Atari 600 z magnetofonem — 140 tys.
ZX 81 — 40—50 tys.

Rozpiętości cen żądanych za ten sam rodzaj sprzętu są bardzo duże i sięgają kilkudziesięciu tysięcy złotych. Sto tysięcy wydaje się jednak być dolną granicą sumy, za jaką można kupić ZX Spectrum 48 kB w komisie.

Powyższe informacje w większości uzyskałem w warszawskim sklepie przy ul. Kijowskiej 11.

Zwracam się do Was w związku z programem na sprawdzenie umiejętności dodawania („Największa przyjemność", BAJTEK nr 1). Proponuję uprościć obsługę programu przez dodanie:

```
9Ø PAUSE 2Ø: PAUSE Ø
1ØØ CLS
11Ø GO TO 1Ø
```

W tym przypadku obsługa ogranicza się tylko do wciśnięcia dowolnego klawisza po operacji komputera; wtedy automatycznie podaje on nowe zmienne.

Włodzimierz Kierns
Białystok

Zgoda. Rzeczywiście, tak uzupełniony program jest prostszy w obsłudze. Co więcej, możemy ten sam (z punktu widzenia użytkownika) efekt uzyskać nieco łatwiej:

```
9Ø PAUSE Ø
1ØØ RUN
```

Słowo kluczowe RUN może być użyte również w tekście programu. Oczywiście autor artykułu — Marek wiedział o tej możliwości. Nie wykorzystał jej ze względu na zupełnie elementarny charakter swego programu. (Tylko dla przedszkolaków!). Zauważamy bowiem, że w obu wersjach „ulepszenia" występują elementy, które wymagały rozszerzenia ram artykułu poza ów „elementarz" — instrukcja skoku albo rekurencyjne wywołanie programu. Mamy też nietrywialne użycie instrukcji PAUSE (na Spec-

trum użyta z argumentem Ø ma nieco inne działanie niż zazwyczaj). Tych rzeczy przedszkolaki z pewnością dowiedzą się niedługo.

☆

Jaka jest cena ZX Spectrum na Zachodzie? Gdzie można się zgłosić, aby nauczyć się pracować na komputerze i poznać Basic? Może istnieją jakieś kluby? Proszę o podanie adresu takiego klubu gdzieś w okolicach Bydgoszczy lub w innym rejonie.

Piotr Buczkowski
ul. Chocimska 7 2
85-078 Bydgoszcz

Cena ZX Spectrum w krajach zachodniej Europy kształtuje się obecnie na poziomie ok. 90 dolarów. RFN jest krajem, w którym ceny interesującego nas sprzętu są najniższe.

Niestety! Nie ma aktualnie żadnego rozeznania co do ilości, rodzaju, lokalizacji i statusu klubów mikroinformatycznych na terenie kraju. Stąd też możemy jedynie zwrócić się z apelem do tego typu placówek: DAJCIE NAM ZNAĆ O SOBIE!

Nie podaję źródeł, w których można znaleźć informacje na temat Basic'u. Po pierwsze dialekty tego języka rozpowszechnione na różnych typach mikrokomputerów różnią się od siebie dosyć znacznie. Po drugie, w żadnym wypadku nie radzę uczenia się tego języka bez kontaktu z komputerem. Będzie to czas w dużym stopniu stracony na przyswojenie sobie wiadomości, których lepiej i skuteczniej uczy praktyka. Opis języka znajdzie Pan na pewno w instrukcji maszyny, przy której Pan zasiądzie.

Poza tym już do pierwszych kontaktów z komputerem gorąco polecam Logo.

☆

Proszę o zamieszczenie na łamach „Bajtka" programów gier komputerowych, np. Atic-Atac, gdyż nie wszyscy mają możliwość zdobycia tego rodzaju programów. Moim zdaniem warto poświęcić na ten cel nawet cały numer „Bajtka".

Czytelnik z Warszawy

W praktyce jest niemożliwością opublikowanie na łamach naszego miesięcznika (dotyczy to także każdego innego czasopisma) tekstu programu rozmiarów Atic-Atac. Pomijając kwestię praw autorskich, programy gier o takim, czy nawet mniejszym stopniu złożoności są w ogromnej większości pisane w języku maszynowym. Rozmiary programu — 30 kB — czynią zadanie druku wykonalnym jedynie w teorii. Któż zresztą z czytelników podjąłby się przepisywania go na swój komputer i poprawiania ewentualnych błędów w takim programie.

Programy publikowane w zagranicznych periodykach, to głównie proste gry napisane w języku Basic. Nie jest wykluczone, że ukażą się i na naszych łamach. Nie możemy natomiast poświęcić na ten cel całej objętości numeru. Proszę pamiętać, że np. gry przeznaczone na Spectrum nie zainteresują posiadaczy komputera Commodore czy TI 99/4A.

Marcin

☆

Z zainteresowaniem przeczytałem dwa pierwsze numery „Bajtka". Niestety, stwierdziłem, że w drukowanych przez Was tekstach programów znalazły się błędy...

Stefan Krajewski
Wrocław

Takie błędy faktycznie w „Bajtku" wystąpiły, do czego przyznajemy się z prawdziwą przykrością. Oto sprostowania autorów.

„OCHRONA PROGRAMÓW" z nru 1: "...Drugi blok danych liczy 110 bajtów, tak więc przy wprowadzeniu bloku "z taśmy" trzeba zmienić linię 30 na

```
3Ø FOR j=1 TO 11
```

Ponadto w linii 26Ø tegoż bloku mylnie wydrukowano jako dziewiątą liczbę na liście danych 31 zamiast 33, w rezultacie czego nie zgadzała się suma kontrolna (na ostatniej pozycji). Linia 1ØØ (uruchomienie programu) winna wyglądać

```
1ØØ RANDOMIZE USR 23760
```

A.K.

"SZANSA", nr 2: „Zakradło się tutaj kilka różnych błędów, w niektórych rozkazach było elementów za dużo, w innych znów za mało, słowem straszny bałagan. Ze wstydem przyznaję, że za drugi odcinek wydruku programu w rubryce „Tylko dla przedszkolaków" należy się dwója. Spieszę więc podać Wam sposób, w jaki można ten program poprawić:

```
8 PRINT' "TWOJE TYPY:"
2Ø INPUT N
3Ø PRINT N; "*";
115 PRINT' "WYLOSOWANO:"
16Ø PRINT' "GIER"; T; "WYDATKI"; T*4Ø
18Ø PRINT' "DZIEKUJE CI BARDZO! WYDATKI:"; T 4Ø
```

Przepraszam Was, Przedszkolaki!

Marek

LOGO: "W pierwszym odcinku na marginesie pominięto kropkę przy omawianiu komend .CONTENTS i .PRIMITIVES.

W pierwszym odcinku niejasno objaśniłem działanie instrukcji SENTENCE — usuwa ona tylko zewnętrzną parę nawiasów. Jeśli jej przedmiotem jest lista list, wówczas listy składowe nadal traktowane są jak listy, a nie jak grupy słów.

Nieco wyżej również w pierwszym odcinku pominięto spacje, co wypaczyło sens wywodu — jako dwie liczby LOGO traktuje wyrażenie 7—5.

W programie na dole str. 13 pierwszego odcinka pominięto natomiast znaki działań arytmetycznych, co pozbawiło go sensu. Powinno być:

```
:TO SZEREG :LICZBA
IF :LICZBA <100 [SZEREG :LICZBA×3]
PR :LICZBA
END
```

Ostatni błąd pojawił się na str. 11 — przesuwanie tekstu na ekranie zatrzymuje SYMBOL SHIFT S, a nie SYMBOL SHIFT Y.

W.M.

COMMODORE C 128

C 128 jest jednym z nowszych produktów amerykańskiej firmy COMMODORE, znanej na naszym rynku głównie jako producent mikrokomputera C 64. Ostatnio trafiają do Polski również nowe modele: C 16 i C 116.



Nowy mikrokomputer różni się od poprzedników wyglądem. Jego profesjonalna klawiatura ma o 28 klawiszy więcej niż C 64 — w tym wyróżniony blok cyfrowy. Nie to stanowi jednak o jego atrakcyjności. Otóż C 128 ma wyodrębnione trzy różne rodzaje pracy (jest wyposażony w trzy mikroprocesory).

Po pierwsze może pracować jako C 64 i wówczas posiada pamięć o pojemności 64 kB RAM i 16 kB ROM oraz wersję Basic'u 2,0. W tym trybie wykorzystywany jest mikroprocesor 6510A z zegarem 1.02 MHz. Dysponujemy obrazem składającym się z 25 linii po 40 znaków w każdej, 16 barwami i 8 „spritami". Rozdzielność ekranu wynosi 320x200 pkt. Oczywiście możemy wykorzystać wszystkie programy napisane na mikrokomputer C 64.

Nowy jest drugi rodzaj pracy: dysponujemy pamięcią o pojemności 128 kB RAM, z możliwością rozszerzenia do 512 kB i nowym Basic'em — wersja 7.0. Język ten posiada ponad 140 rozkazów (wszystkie rozkazy poprzednich wersji plus zupełnie nowe). Możemy wybierać pomiędzy obrazem o 25 liniach i 40 kolumnach a obrazem o 80 kolumnach (640 x 200 pkt.). To wszystko w 16 kolorach. W tym systemie pracuje mikroprocesor 850Z (jego lista rozkazów jest kompatybilna z 650Z)

Wreszcie trzeci rodzaj pracy to CP/M 3.0 z mikroprocesorem Z80A, dzięki któremu C 128 staje się urządzeniem w pełni profesjonalnym.

Nowe są również peryferia. Stacja dysków C 1571 ma dyskietki 5 i 1/4 calowe. Pojemność jednej dyskietki wynosi 350 kB. Stacja dysków sterowana jest własnym mikroprocesorem 650Z (32 kB ROM i 2 kB RAM).

Szybkość zapisu wynosi odpowiednio:
300 bodów w systemie C 64
1500 bodów w systemie C 128
3500 bodów w systemie CP/M

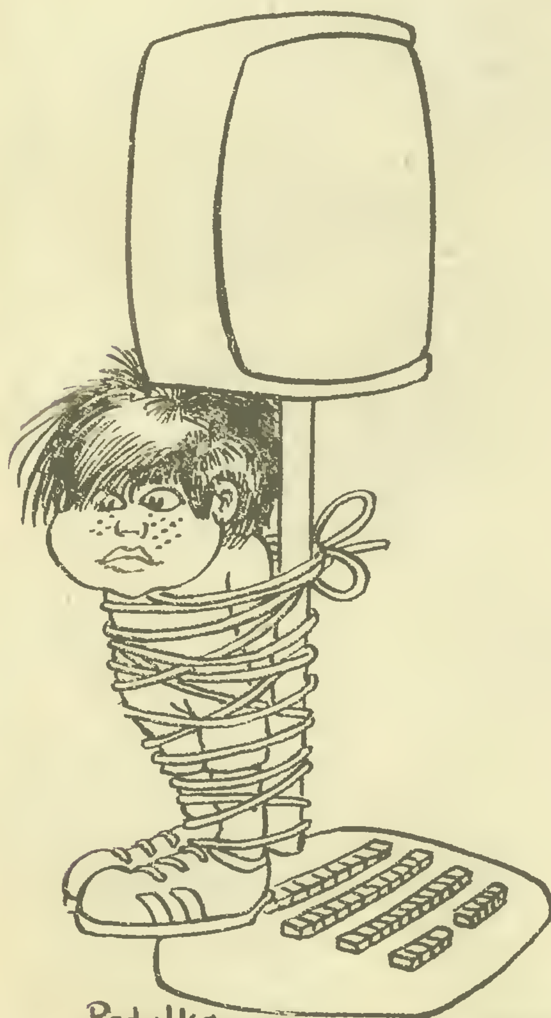
Dużą zaletą C 128 z naszego punktu widzenia jest możliwość wykorzystania dość już licznych w kraju programów na C 64 z równoczesnym ogromnym poszerzeniem możliwości mikrokomputera. Jedyną, moim zdaniem, wadą tego urządzenia jest jego cena — około 1200 marek. Stacja dysków C 1571 kosztuje około 1000 marek.

Mikroprocesor: 6510A; 850Z; Z80A
System pracy: Basic V 2.0; Basic V 7.0; CP/M
Pojemność pamięci: 128 kB (możliwość rozszerzenia do 512 kB)
Pamięć masowa: magnetofon kasetowy, 5, 1/4 Floppy
Kolor: 16 barw
Ilość znaków w linii: 40 lub 80
Ilość linii: 25
Dźwięk: 6581 SID Chip
Klawiatura: maszynowa z wyodrębnionym blokiem cyfrowym i klawiszami funkcyjnymi
Przyłącza:
— dwa gniazdka dla joysticków (drążków sterowych)
— User Port
— rozszerzenie ROM.

KRZYSZTOF KRUPA

WIĘCEJ RUCHU!

**Przypuszczam, że i Was denerwuje brak ruchu na ekranie. A bez ruchu — wiadomo, nie można prawie w nic grać i znacznie szybciej nudzi się nam siedzenie przed telewizorem.
Jedyna rada —
rozruszać znaki na ekranie.**



Określimy sobie najpierw, jaką literkę (albo znak graficzny) uruchomimy. W naszym przykładzie będzie to „B". po prostu z sympatii dla BAJTKA. Po ustawieniu litery w wymaganym miejscu zaczynamy. Ekran w komputeracii podzielony jest na pewną ilosc wierszy, z których każdy może zawierać określoną ilość znaków. Najpopularniejsze w Polsce komputery ZX Spectrum dzielą ekran telewizora na 22 wiersze (numerowane od 0 do 21), w każdym wierszu jest możliwość zapisania 32 znaków (0 do 31). Umiejscawiając znak podajemy najpierw numer wiersza a następnie numer miejsca znaku.

Każmy więc naszemu komputerowi drukować litery w coraz to wyższych wierszach, przy zastosowaniu instrukcji FOR. Rozkazy 10 20 i 50 spowodują wydrukowanie od dołu do góry kolumienki litery B. Działanie to spowolnimy przez instrukcję PAUSE rozkaz 30. Wykonując ją, maszyna czeka przez pewien czas w trakcie wykonywania tego bloku programu. Zwiększając liczbę określającą pauzę, wydłużamy czas jej trwania.

Staramy się jednak uruchomić punkt (literę), a nie drukować kolumnę. Trzeba więc skasować wyświetloną literę, rozkaz 40, wydrukowac nową w innym miejscu, skasować i tak dalej, uzyskując wrażenie ruchu.

Zmieniając odpowiednio rozkazy 20 i 40, możemy uzyskać ruch w innych kierunkach, a więc w pionie z góry na dół, w poziomie z lewej do prawej i odwrotnie, a nawet, kiedy trochę pomyślimy i ruchy na ukos ekranu. Przy przesuwaniu poziomym literka zajmuje miejsca w wierszu od 0 do 31 (z lewej do prawej) lub od 31 do 0 (przypominam STEP — 1) z prawa do lewa. No a ukos, rzecz prosta, w każdym kroku maszyny musi realizowac przesunięcie w pionie i w poziomie.

```
1Ø FOR X=21 TO Ø STEP — 1
2Ø PRINT AT X, 16; "B"
3Ø PAUSE 1Ø
4Ø PRINT AT X, 16; " "
5Ø NEXT X
6Ø PRINT "KONIEC"
```

W ten sposób wprowadziliśmy w życie tytułowe hasło, ale jeszcze niezupełnie. Proponuję więc teraz: wyłączmy całą maszynerię i biegiem na spacer. Najważniejsze bowiem to:

WIĘCEJ RUCHU!

MAREK



```
10 LET A = .4: LET B = .02:
   LET C = .5
20 FOR i = 0 TO 174 (1 +C)
30 CIRCLE 125 + A × i × SIN (B × i) ,i,
C × i
40 NEXT i
```



```
10 LET A = .8: LET B = .08:
   LET C = .08
20 FOR i = 0 TO 174 (1 +C)
30 CIRCLE 125 + A × i × SIN (B × i) ,i,
C × i
40 NEXT i
```



```
10 LET A = .8: LET B = .045:
   LET C = .25
20 FOR i = 0 TO 174/(1 +C)
30 CIRCLE 125 + A × i × SIN (B × i) ,i,
C × i
40 NEXT i
```



```
10 LET A = .5: LET B = [illegible]
   LET C = 1
20 FOR i = 0 TO 174 (1 +C)
30 CIRCLE 125 + A × i × SIN (B × i) ,i,
C × i
40 NEXT i
```

Prezentowany przykład wykazuje, jak interesujące efekty uzyskać można przy pomocy bardzo krótkich programów graficznych. W naszym przypadku są to zaledwie trzy linie.

Zwróćcie uwagę na to, że wszystkie grafiki zostały narysowane w wyniku realizacji tego samego programu — zmieniły się jedynie parametry.